A Packing Approach for the Early Stage Design of Service Vessels

A Packing Approach for the Early Stage Design of Service Vessels

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universtiteit Delft;
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben;
voorzitter van het College voor Promoties
in het openbaar te verdedigen

op maandag 12 december 2011 om 12:30 uur

door

Bartholomeus Johannes VAN OERS

Scheepsbouwkundig ingenieur
geboren te Breda

Dit proefschrift is goedgekeurd door de promotoren:

Prof.ir. D. Stapersma
Prof.ir. J.J. Hopman

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.ir. D. Stapersma	Technische Universiteit Delft, promotor
Prof.ir. J.J. Hopman	Technische Universiteit Delft, promotor
Prof.dr. S.O. Erikstad	Norwegian University of Science and Technology
Prof. D.J. Andrews*	University College London
	**FREng PhD FRINA FIMechE RCNC*
Prof.dr. F.M. Brazier	Technische Universiteit Delft
Prof.dr. C. Witteveen	Technische Universiteit Delft
Dr. D.J. Singer	University of Michigan
Prof.dr. T. Tomiyama	Technische Universiteit Delft, reservelid

Published by

VSSD
internet: http://www.vssd.nl/hlf
e-mail: hlf@vssd.nl

keywords: acceptance, arrangement, compromise decision, configuration, drillship, engineering judgement, feasible ship design, incomplete set of objectives, layout, naval architecture, optimisation algorithm, packing problem, parametric ship description, selection problem, service vessel, ship design, too optimal, trade-off, transparency, warship

ISBN 978-90-6562-283-9

To Léonie, Isolde and my soon-to-be-born son

Contents

Preface

The thesis you find before you is the result of a research project that strives to improve the early stage design process of service vessels, a class of complex ships that perform demanding operations at sea.

The complexity of designing such ships is illustrated by two quotes. Though both quotes deal with warship design, I consider the issues they raise to be universal and deem them applicable to ship design in general and to the design of service vessels in particular.

A warship is engineering's greatest compromise (quoted from *Purvis [102]*).

The first quote, by M.K. Purvis, a Deputy Director of Naval Construction at the Ministry of Defence in the United Kingdom, exemplifies a common held belief among naval architects, who tend to say that 'every ship is a compromise'.

You cannot have everything. If you attempt it, you will lose everything. On a given tonnage, there cannot be the highest speed ***and*** *the heaviest battery* ***and*** *the thickest armour,* ***and*** *longest coal endurance* (from *Mahan [74]*, quoted in *Hughes [61]*).

The second quote, by Alfred Mahan, the US naval strategist, illustrates that a ship design is not just any compromise, but the particular compromise that the naval architect chooses to create. When designing the battleship referred to in the quote by Mahan, a naval architect must consider benefits and drawbacks of higher speed versus better weapons and longer endurance versus thicker armour, before deciding which compromise meets the owner's needs best. The same need for compromise arises when designing service vessels in the present day, regardless of whether it is a deep water drilling vessel, a dredger, or a modern warship such as a frigate or an amphibious assault ship.

Traditionally, naval architects use compromise decisions as the starting point for the design process. They then use them to create a ship design, predicts its performances, and study whether the desired compromise can be achieved. All too often, it cannot be, because the particular compromise turns out to be impossible, unpractical or unaffordable. Moreover, insight gained during the design process might cause the owner to adjust his opinion on the desirability of the compromise that served as a starting point.

Hence, the traditional approach has two drawbacks. Firstly, the compromise used at the start may not be the compromise the owner wants at the end, and secondly, it might not be possible to create a design that reflects the compromise used at the start.

The approach presented in this thesis takes the opposite route in order to remove both drawbacks and, thereby, improve the design process of complex service vessels. It does so by first generating a large number of feasible, three-dimensional ship designs to cover a broad range of potentially interesting compromises, and second, by helping the naval architect to study them thoroughly to see which designs represent the most desirable compromise.

The research was carried out from 2005 to 2009 as a full-time PhD project at Delft University of Technology under the supervision of prof. ir. Douwe Stapersma, prof. dr. ir. Tom van Terwisga (until October 2006) and prof. ir. Hans Hopman (from October 2006 onwards).

The project was funded by the Royal Netherlands Naval College (now the Netherlands Defence Academy) and carried out in close collaboration with the Defence Materiel Organisation of the Netherlands Ministry of Defence. Several other organisations, e.g., the Maritime Research Institute The Netherlands, NEVESBU and GustoMSc also provided support in the form of software, access to design data and supervision of students.

A glossary with definitions is provided in Appendix A to help out with the terminology that is used throughout the text.

I hope you find the thesis of interest and welcome any comments, suggestions or questions you may have. You can send them to the email address below.

Bart van Oers
The Hague, The Netherlands
`comments_on_packing_based_ship_design@xs4all.nl`

Chapter 1

Introduction

1.1 Introduction: service vessels

Fig. 1.1 – Two transport vessels: an oil tanker (upper-left) and car ferry (lower-left) and two service vessels: a frigate (upper-right) and a deep water drillship (lower-right)

Thousands of ships and other floating structures ply the oceans every day to serve humanity in a variety of ways. This large fleet can be divided into two classes of ships: transport vessels and service vessels. The former transport cargo between two places; well-known examples include oil tankers, bulk carriers and LNG-tankers as well as car ferries (Fig. 1.1 shows two examples). The latter, service vessels, use their 'cargo' to perform missions at sea. Common examples include warships, e.g., frigates, amphibious assault ships and aircraft carriers, as well as civilian ships, e.g., cruise ships[1], drillships, crane vessels, pipe-laying vessels and dredgers (again, Fig. 1.1 shows two examples). The mission obviously differs per type of service vessel. Common examples include drilling for hydrocarbons, installing oil platforms, reclaiming land, carrying out maritime security operations, providing air-defence or hunting submarines.

Over the past decades, European shipyards and design consultancies have specialised successfully in the design and production of both military and civilian service vessels. The reason is simple: to weather competition from lower-wage countries with a competitive advantage in building the -often simpler- transport vessels. This specialisation, however, remains successful only by the shipyards' and consultancies' continuing ability to design ever-more-competitive service vessels. The research presented in this dissertation aims to sustain and improve this ability, by improving the process used for the early stage design of service vessels. This requires a discussion on the design of service vessels, which Section 1.2 provides.

1.2 Designing service vessels: finding coherent configurations of relevant systems with suitable performances

The missions performed by service vessels typically require large and complex systems. Systems used by civilian vessels include a drilling derrick to drill for hydrocarbons, a large crane for offshore lifting, and a suction pipe, dredging pumps and a hopper to dredge (see Fig. 1.2 for some examples). Systems used by military service vessels include radars or other sensors, weapons, and a helicopter flight deck and hangar (again, see Fig. 1.2).

Their sheer size and their complexity ensure that the systems have a major impact on the design of a service vessel. This impact affects the design in three specific ways. Therefore, a service vessel with at least a modicum of utility has to have the following related features.

- First, the ship has to have the **relevant systems** aboard to carry out her mission. A functional decomposition such as discussed in *Andrews et al. [10]*, *Wolff [140]* and *Klein Woud and Stapersma [69]* can help establish the relevant systems.

  Furthermore, choosing systems has follow-on consequences, as systems often need support provided by other systems.For example, a radar needs chilled water and electrical power provided by chillers and diesel generators.

  The size of systems means that the mere requirement to carry them has a considerable impact on ship size. Still, carrying them is not sufficient; otherwise a simple barge with the systems stacked on-top would suffice.

- Second, a service vessel must be able to use its systems to perform the mission. This requires proper integration of the systems into a **coherent ship configuration** to ensure all systems work together correctly. For example, a radar needs both an unobstructed field of view and uncluttered access for the maintenance crew.

  The term configuration is used in this dissertation to describe the relative positions of systems. Reason is that the common term 'arrangement' typically concerns the internal layout of equipment inside a fixed envelope (which consists of hull and superstructure),

[1]Cruise ships are considered to be service vessels as they entertain their passengers during their voyage.

Fig. 1.2 – Examples of systems used by service vessels

whereas the research presented in this dissertation will look into the variation of envelope shape and size concurrently with the variation of the arrangement of systems.

- Third, the ship as a whole must have **suitable performance levels** to ensure that the systems can operate in the required environment, which typically requires sufficient stability and speed, good sea-keeping and manoeuvrability. Lower performance could cause the service vessel to be less competitive. For example, a crane vessel with a low transit speed might be too slow to travel between two job locations in the required time, reducing the number of lifts it can do in a season. Military vessels need suitable military performance levels, e.g., the ability to deploy weapons effectively, as well as proper levels of susceptibility, vulnerability and recoverability, in order to succeed in the man-made hostile environment in which the ship carries out her mission. One special performance of interest to both military and civilian service vessels is cost (which is considered a performance because it results from the ship's design).

The three features imply that designing a service vessel entails finding a configuration of relevant systems that work correctly by being properly integrated in a well performing ship.

Naval architects use the process shown in Fig. 1.3 to guide their design effort towards suitable ship designs. It shows the so-called systems engineering 'V-diagram' which lists the steps of the design process and their dependencies.

These steps take a naval architect from establishing the ship's mission (what should be achieved), through developing the functions that must be fulfilled to complete the mission successfully (how it will be achieved), to choosing the systems that will be integrated in the ship (which, with the crew, offer the means to achieve the mission). These systems (themselves consisting of subsystems and components) are then integrated in an actual ship by creating a configuration, which serves as input for performance predictions. With the performances estimated, the extent with which each of the functions is fulfilled can be established. Lastly, the results of the different performance predictions are combined in an overall measure of the ship's ability to perform the mission: her mission effectiveness.

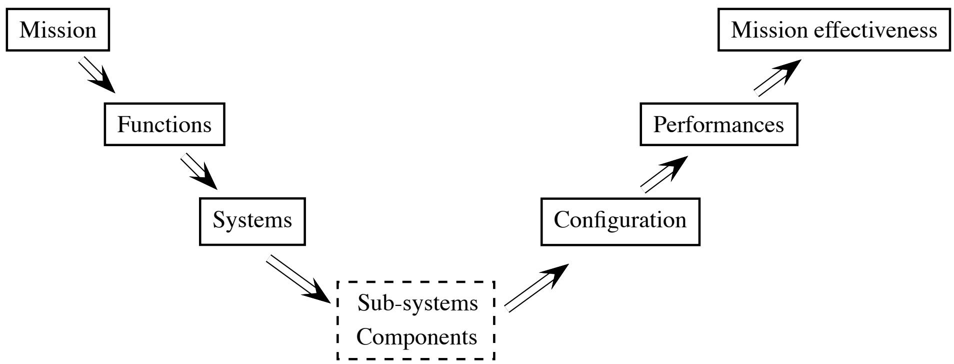


Fig. 1.3 – From mission via systems, configuration, performance to mission effectiveness

Three remarks are in order with regard to applying the steps in Fig. 1.3. First, typically a distinction is made between systems, subsystems and components. For this reason, subsystems and components have been included in Fig. 1.3. However, subsystems and component design and selection will not be considered in this dissertation (hence, the dashed box), as early stage ship design typically does not reach this level of detail. Instead, this thesis will use the term 'systems' to define all parts of the ship's design in a generic manner, regardless of their level of detail. The term 'systems' will therefore also be used to describe subsystems and components. Second, not all steps in Fig. 1.3 may need or get the same amount of attention in every design project. For example, when designing a service vessel that will be a sister ship of an existing vessel, one will not focus on formulating the mission and functions again. Instead, one will focus on improving particular systems and adjusting the configuration, e.g., to reduce production cost. Third, feedback between the different steps in Fig. 1.3 is discussed in Section 1.4.

This Chapter uses the diagram in Fig. 1.3 to highlight various aspects of the design of service vessels and the process used to design them. Therefore, all steps in Fig. 1.3 are of interest, but this does not mean they are of equal importance.

Two of the steps in Fig. 1.3: the choice of systems and configuration of these systems, are of particular interest during the design of service vessels. Therefore, both should be considered from the earliest stage of the design process, as proposed by *Andrews [3]* and *Andrews et al. [10]*. The reason for the former is obvious, doing otherwise will equip the vessel with the wrong systems for the missions she is built to perform. The reason for the latter is more complex. Configuration is important because only placing all systems at their actual positions in the ship will reveal whether they fit and work correctly (as argued by *Andrews [3]*). Moreover, only when the complete configuration is available at a suitable level of detail can performances, such as sea-keeping or vulnerability, be predicted (as discussed in *Van Oers and Stapersma [128]*).

The selection of relevant systems and their integration in a suitable configuration, challenging enough in itself, is complicated by the many options the naval architect has to consider. Section 1.3 discusses them.

1.3 Many options to consider

The naval architect is faced with a large number of options when designing a service vessel. The options concern the three features introduced previously: systems, configuration and performances (see Section 1.2).

These options are important, as they give the naval architect the chance to improve the competitiveness of the ship by choosing the right combination of systems, configuration and performance levels. Therefore, their proper investigation warrants considerable attention.

The options are discussed below.

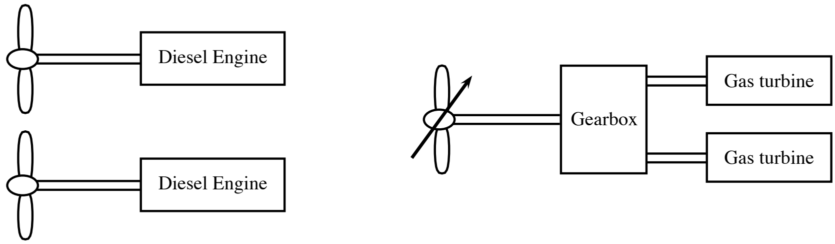


Fig. 1.4 – Different propulsion system options: type, number and connectivity

- **Systems**. The options one must consider concern system type, number and connectivity, as well as the resulting shape, size, weight and other characteristics (e.g., fuel consumption or required cooling capacity). Fig. 1.4 shows two examples with different system types, number and connectivity. The type of system is relevant because different systems exist that may fulfil the same function. For example, a crane, davits and a well dock, can all be used to launch and recover small landing craft. Note, hull and superstructure are treated as systems, just like a fire-fighting pump or a fuel tank. The number of systems is also important, e.g., one can choose to use either two or four cranes to transfer tubulars from a transport barge onto a pipe-laying vessel. Connectivity deals with how systems are connected. The connections of chillers in a chilled water distribution system can differ due to changes in required redundancy. When the type of system is chosen, its shape, size and weight are also determined.

- **Configuration**. The options deal with the different positions systems can have relative to each other (see Fig. 1.5 for two simple examples). For instance, an aircraft hangar can be located either adjacent, or underneath a flight deck, with each position having its own merits. Choosing different system positions can have a considerable impact on attained performance levels, e.g., wave-induced acceleration (see *Keuning [66]* for an example).

- **Performances**. The options one must consider are the required performance levels the ship must achieve (see Fig. 1.6 for two simple, stability-related examples). For example, performing anti-submarine warfare around a convoy of civilian vessels might be possible with a maximum speed of 25 knots, while protecting an aircraft carrier might require maximum speeds of over 30 knots. Different cost should also be considered, depending on the flexibility of the budget an owner is willing to spend. Cost is of special importance for warships, which generally must adhere to a fixed procurement budget.

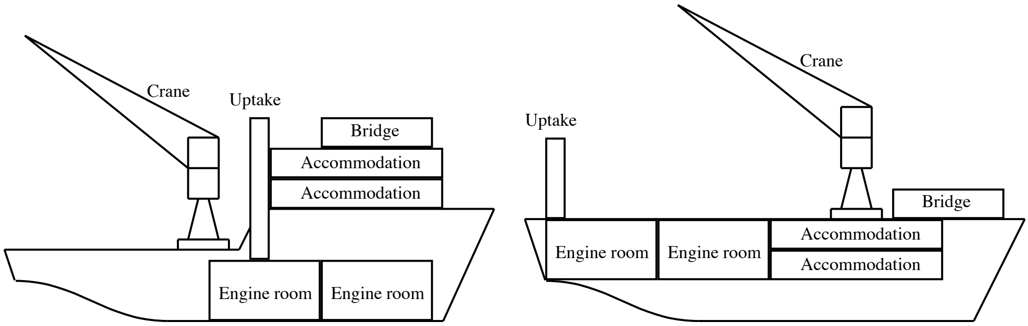


Fig. 1.5 – Two different configurations of the same set of systems

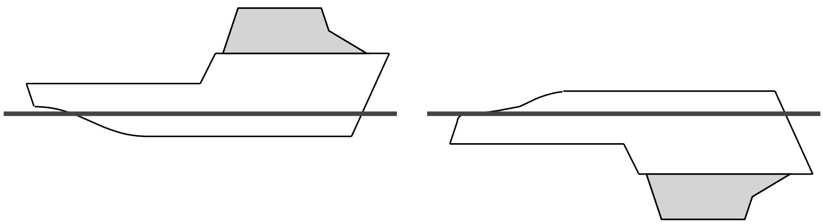


Fig. 1.6 – Two different performance levels: stable and unstable

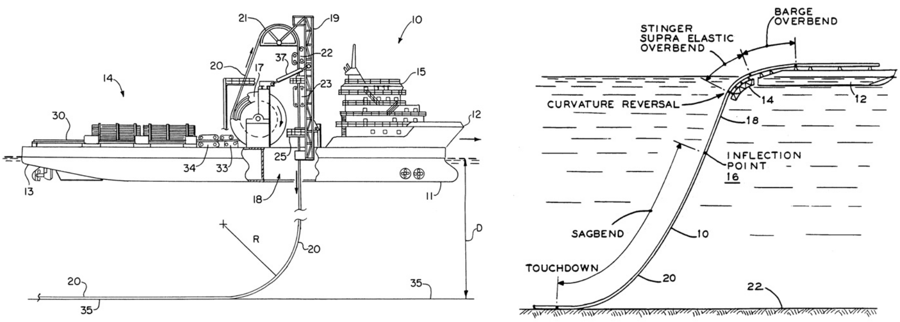


Fig. 1.7 – Two off-shore pipe-laying approaches: so-called J-Lay (left) and S-Lay (right)

Choices made for one feature obviously influence the choices for other features (see Section 1.4). For example, a pipe-laying vessel with a so-called ‘J-lay’ system has completely different systems and, therefore, a different configuration than one with an ‘S-lay’ system (see Fig. 1.7). *Graham [50]* provides a discussion of such influences in warship design. Needless to say, these influences further complicate the design of service vessels, but handling them properly also offers a change to improve their competitiveness.

The large number of options with respect to systems choice, number and connectivity, as well as the many alternative configurations systems can have, and different performance levels one may require, give rise to a large number of alternative ship designs. Still, not all alternatives are equally useful, and Section 1.4 discusses why.

1.4 Design requirements

Service vessels are designed, built and operated to realise a perceived future utility. Civilian ships are built to pursue economic advantage, i.e., profit, while building and operating warships offer a country the means to realise its political ambitions.

The utility of a future ship is usually not expressed in terms relevant to the naval architect. Instead, the owner -frequently in collaboration with naval architects- translates the perceived, future utility into a set of design requirements, which prescribe the technical abilities a future ship should have. Attaining these requirements means the future ship should be competitive enough to realise the utility perceived at the start of the design process. As such, design requirements are largely -but not exclusively- derived from the ship's mission and the functions she is to perform. Also, not all design requirements will be stated formally at the start of the design process. Often, additional requirements will appear once more insight is gained by creating one or more ship designs.

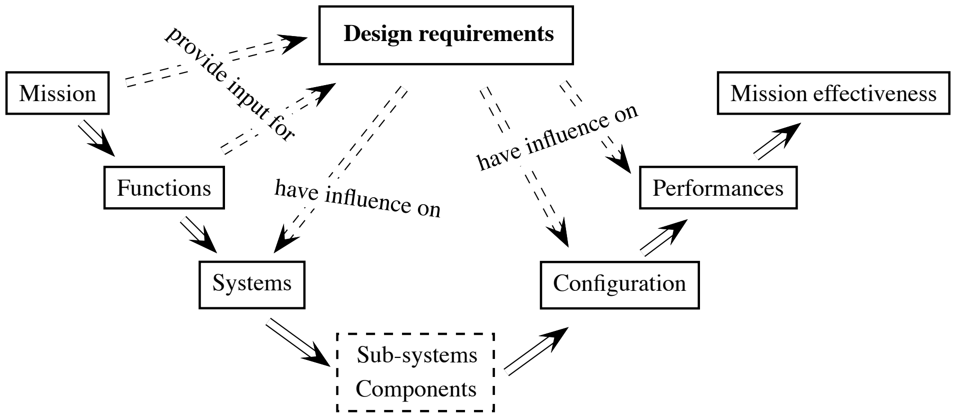


Fig. 1.8 – Influence of design requirements

Design requirements may concern any of the three features discussed in Sections 1.2 and 1.3: which systems, what configuration or what performance level a service vessel should have (shown in Fig. 1.8). The first type of requirement could specify the type of drilling derrick, for instance. The second type may state the separation of redundant diesel generators needed to safeguard dynamic positioning ability with one compartment flooded or on fire. The third type can state required maximum speed or helicopter operability in a particular sea state. Fig. 1.8 also shows the relation between mission, functions and design requirements.

Functional requirements that state what a ship should be able to do, e.g., 'provide air defence', are not used in this dissertation (see *Wolff [140]* for a thorough discussion on functional requirements and their complexity). The reason is that function requirements can only be fulfilled by choosing specific systems and attaining particular performance levels. Prescribing these required systems and required performance levels can be achieved using the three types of requirements described earlier, which removes the need to use dedicated functional requirements.

The design of a service vessel must obviously meet the stated design requirements, in order to be acceptable to an owner. This means design requirements can be used to distinguish

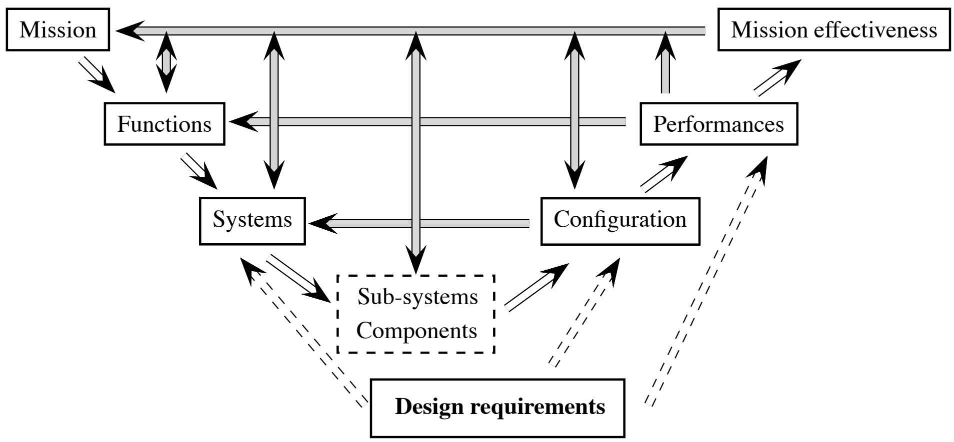


Fig. 1.9 – Feedback to handle interactions between design requirements integrated in the steps of Fig. 1.3

between the many alternatives that exist (Section 1.3). The basis for distinguishing them is the observation that not all design requirements are considered equal. Two classes of requirements are identified, based on their negotiability.

- Non-negotiable requirements. Some requirements are non-negotiable and are called feasibility requirements in this dissertation. The reason is that failure to comply with such requirements affects the physical, financial or political feasibility of the vessel. A typical example of a non-negotiable requirement is the need for a surface vessel to float in stable, upright condition. A ship design that complies with all stated non-negotiable requirements is called feasible in this thesis.

- Negotiable requirements. Negotiable design requirements can be adjusted -within limits- to attain or improve other non-negotiable or negotiable requirements. One example is the speed of a warship, which could be reduced to reduce procurement cost, e.g., as used for the Royal Netherlands Navy Holland class patrol ships. For other warships though, speed cannot be reduced, e.g., a frigate built to escort an aircraft carrier has to meet a non-negotiable minimum speed requirement to be able to keep up with the carrier battle group.

The distinction between negotiable and non-negotiable is complex and may change significantly between different design projects. Ultimately, it is decided upon by the owner and / or naval architect. In practice, however, one can assume that some requirements are always non-negotiable, such as the ability of a surface ship to float in stable, upright condition. Such differences in negotiability are exploited in Chapter 3.

Most importantly, design requirements cannot be formulated independent from one another, regardless of their negotiability. The integrated nature of a ship's design (see *Andrews [3]* and *Van der Nat [120]*) means that stating and meeting one design requirement may affect the ability to attain other requirements. Therefore, one needs feedback between the steps in Fig. 1.3 to identify and resolve such interactions between requirements (as shown in Fig. 1.9).

A simple example of such interactions is an increase in required cruise speed, which through an increase in resistance will lead to increased fuel consumption, which in turn affects range. Without an increase in fuel tank capacity, the ship will fail to meet range requirements. Increasing fuel capacity, however, could interfere with a requirement on maximum displacement. Some combinations of design requirements may conflict as a result of the integrated nature of ship design. In that case, it necessitates a compromise by trading off one negotiable requirement against other requirements. Hence, some combinations of requirements can be impossible to achieve due to the governing laws of physics. *Mayer and Cooper [78]* discussed the physical impossibility to meet a 40 knots speed requirement with a Landing Platform Dock ship; only a speed of 32 knots could be achieved after an extensive research and design effort.

In summary, the early stage design of a service vessel can be characterised as finding a configuration of relevant systems that forms a suitable compromise between several negotiable design requirements and that complies with all non-negotiable requirements. Finding such a design among the available alternatives is not straightforward, as Section 1.5 explains.

1.5 Some problems encountered in the early stage design of service vessels

The Sections 1.1 and 1.2 introduced service vessels and discussed their desirable features with respect to their systems, configuration and performances. Section 1.3 subsequently covered the many options a naval architect can and should consider during the early stage design of these vessels, while Section 1.4 discussed the requirements the future ship must comply with in order to be of value to its owner, regardless of the particular combination of options it uses. Together, these four Sections provide the background needed to introduce the problems this dissertation will address.

Section 1.4 characterised the early stage design of a service vessel as a search, among the many alternatives, for a configuration of relevant systems, which complies with all non-negotiable requirements, and forms a suitable compromise between several negotiable design requirements. These alternatives, crucially, offer the naval architect the chance to tailor the ship's design so that it best meets the design requirements and, therefore, will be competitive.

Finding such designs requires a thorough investigation of the options to see which alternatives are promising. Unfortunately, such an investigation is cumbersome to carry out for three related reasons.

- First, evaluating whether a particular alternative complies with the design requirements requires performing the steps in Fig. 1.3. This means one should -at the very least- choose systems, create a ship configuration at a suitable level of detail, and predict performances, before a comparison with the design requirements is possible.

  This requires considerable human effort, as systems, configurations and performance predictions all need to be designed, either prior to the design process or during the design process (see Fig. 1.10).

- Second, one must repeat the steps from Fig. 1.3 for every alternative one wants to check compliance with design requirements for. This constitutes a tremendous effort, given the large number of potentially interesting options that exist (following Section 1.3).

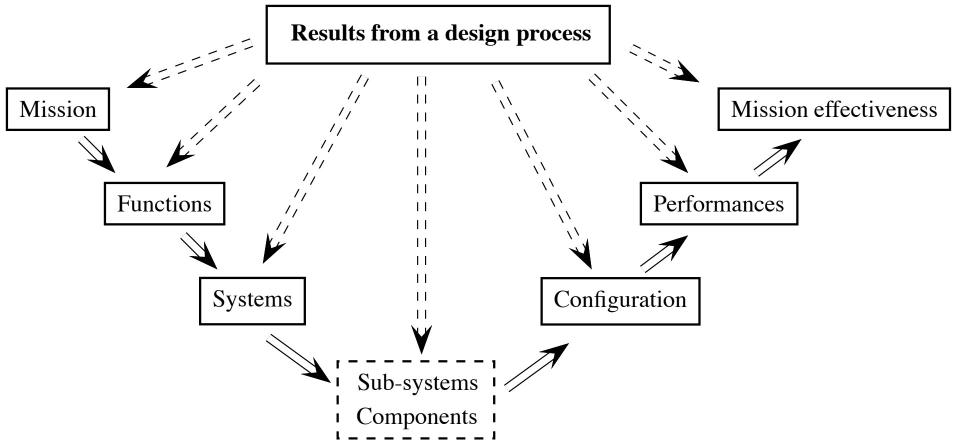


Fig. 1.10 – Designing is everywhere

- Third, choosing systems and creating alternative configurations is relatively straightforward if the design effort focuses on meeting a static set of design requirements. Unfortunately, creating configurations will provide insight in the suitability of design requirements (termed 'requirement elucidation' by *Andrews [5]*). Such insight may cause changes in the design requirements. Changes, which led *Andrews [4]* to use the term 'wicked problem[2]' (see *Rittel and Webber [106]* for a discussion) to describe the complexity that results from the mutual dependency between a ship's design and its underlying requirements. One cause of this mutual dependency is the fixed procurement budget warships have to adhere to, which forces requirements to change to meet the available budget.

  These changes in design requirements mean the sequential adaptation of a single configuration (i.e., following the 'design spiral', as discussed in *Buxton [24]*) will give problems to accommodate the frequent changes to the ship's design. More importantly, alternatives may start to diverge, i.e., differ so much, that a completely new design is necessary to accommodate all changes in design requirements, which further increases the effort required to evaluate the suitability of these alternatives.

In summary, investigating the range of alternatives means that the naval architect must generate a configuration of systems (at an appropriate level of detail) and predict its performances, for every interesting combination of options.

Unfortunately, this -currently- requires so much effort that it is difficult to carry out in the limited time available during the early stage design of service vessels. This means it hampers the naval architect when he / she tries to find a suitable configuration of relevant systems that forms an acceptable compromise between negotiable requirements, and that also complies with the non-negotiable requirements.

Still, one might wonder whether this problem matters. It does, as the large effort required to investigate many alternatives means naval architects will consider only a few alternatives and, hence, have to resort to assumptions, rules of thumb and other short-cuts that both limit the

[2] A ship design problem is considered 'wicked' by *Andrews [5]* if formulating the requirements is more complex than designing the ship that meets them.

ability to design more competitive service vessels and increase risk to ship design firms and shipyards alike. In short, the inability to investigate a large number of alternatives is considered to make it more difficult for the naval architect to design a competitive service vessel.

One may be tempted to think that a relatively simple approach exists that could address the problems outlined above. However, several obvious approaches exist that appear to address these issues but fail to do so in practice.

- Solely using a numerical description of the configuration (e.g., available and consumed deck area, as in *Keizer [64]* and *Van Es and Van Hees [123]*) fails to capture the complexities of the spatial integration of large systems (as argued by *Andrews [3]*).

- Reducing the number of alternatives might work, provided a sufficient number of designs remains to cover the most interesting alternatives. This places a considerable burden on the naval architect to identify those interesting alternatives upfront. This can be notoriously difficult if one designs a vessel with an unfamiliar configuration or with unfamiliar systems.

- Developing only part of the configuration helps to avoid spending effort on alternatives that will be discarded. Still, it does not help to generate alternatives that meet all non-negotiable requirements and that forms a promising compromise between negotiable requirements. To do this, one must develop the entire configuration for a considerable number of alternatives at an appropriate level of detail.

- Creating an entire configuration, but with a lower level of detail, also reduces effort, but again may not capture the complexities of the integration of the large systems aboard service vessels. Moreover, both *Van der Nat [120]* and *Andrews [5]* discussed the need to address ship design problems at an appropriate level of detail. *Andrews [5]*, for instance, discussed the need to carry out detailed damaged stability predictions during the early stage design of a future Royal Fleet Auxiliary tanker to investigate the impact of new regulations that prescribed the use of a so-called ‘double hull’ that limits pollution in case of grounding or collision. This means the required level of detail is difficult to determine beforehand, as it depends on the design problem at hand. It even may necessitate the use of detailed ship configurations very early on in the design process (see *Andrews [5]*).

Summarised, support is necessary to reduce the effort required to thoroughly investigate the many options a naval architect faces. Also, the unsuitability of these simpler approaches means an alternative, more complex design approach is required. Before developing it, however, Section 1.6 will first discuss the focus of the research effort presented in this dissertation.

1.6 Focus of the research

A last issue warrants attention before starting to the development of the novel design approach. Should reducing the effort required to investigate numerous alternatives be considered of particular importance, or are there other worthwhile avenues to improve the early stage design of service vessels?

After all, naval architects face numerous other problems during early stage ship design. For example, developing more accurate prediction tools could be regarded equally important (e.g., as by *Tuitman [116]*), as is the proper formulation of the ship's mission and its decomposition into a coherent set of functions (e.g., as by *Wolff [140]* and by *Brouwer [19]*). In addition, improving the systems that enable the ship to perform her missions is also important. Given these alternative problems to investigate, a decision on the focus of the research is required.

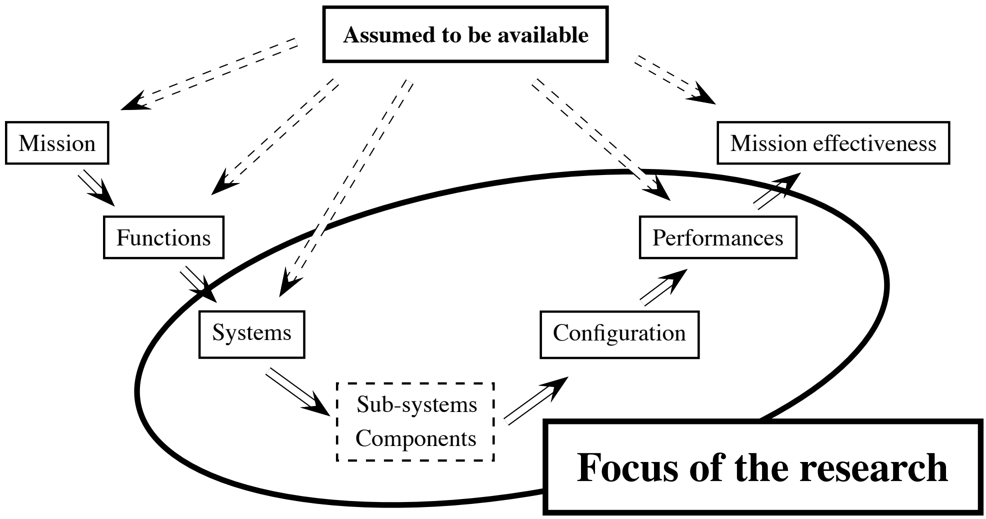


Fig. 1.11 – Focus of the research in this dissertation

The research presented in this dissertation will focus on reducing the effort required to investigate numerous alternative configurations during early stage design, instead of, for example, improving performance prediction tools. More specifically, it will try to do so by improving the speed with which existing (i.e., known) systems can be arranged into a coherent and feasible ship configuration that serves as input for existing prediction tools (as shown in Fig. 1.11).

The reasons to focus on the arrangement process of systems are twofold. Both relate to the fact that the configuration is the link between the systems that -together- enable the ship to carry out her missions, and the performance levels required to use the systems at sea.

- First, being able to both develop and choose relevant systems from the ship's mission (and determine their characteristics) is crucial for the successful outcome of the design process. Still, it is not sufficient just to create an appropriate ship design. To assess whether these systems can be used at sea to perform the mission, one must be able to predict performances. Predicting these performances, in turn, requires the ability to generate configurations.

- Second, improving prediction tools is only useful if, first, the input (i.e., the systems and their configuration) is available to use them in practice, and, second, the input is flexible enough to undergo the large changes commonly experienced in early stage ship design (see *Van Oers and Stapersma [128]*). Both availability and flexibility can be achieved by increasing the speed with which one can generate and alter configurations. This makes improving the accuracy of predictions more worthwhile, as the resulting improvements in predictive power can be exploited to improve the ship's design, instead of serving to validate the assumptions on which the design is based.

Choosing to focus on improving the arrangement process of systems also necessitates several assumptions (as shown in Fig. 1.11). They are discussed below.

- First, it assumes the ship's mission has been defined, as have the functions describing how it will be carried out. Note that changes in systems could be used to change the underlying functions indirectly.
- Second, it assumes that both the systems are available, and their characteristics are known. This assumption is regarded as acceptable, as in general, systems are developed during research and development in separate efforts prior to early stage ship design. In some particular cases, e.g., warships, system design and early stage ship design are performed concurrently.
- Third, it assumes systems require little to no attention or alteration during the early stage ship design process, i.e., their characteristics should be relatively independent on the ship configuration at hand. Though valid for most systems (*Andrews and Dicks [7]* provide a brief overview), this assumption is particularly troublesome for marine engineering related systems (e.g., those providing propulsion power, electricity, chilled water and air conditioning), as well as for ballast tanks, which are used to compensate for changes in variable loads. For such systems, rules will be developed to account for their impact and dependency on the ship's design (similar to the sizing rules used by *Van der Nat [120]*).
- Fourth, it assumes performance prediction tools are available and that they require little to no attention or alteration during the early stage ship design process (like systems). Again, this assumption is regarded as acceptable, as performance prediction tools are developed during research efforts prior to a ship design project (occasionally, prediction tools are modified to be able to predict performances of ship designs not foreseen during their original development).

With the focus settled and the assumptions laid out, Section 1.7 can now formally start the development of the novel design approach by stating the research objective.

1.7 The research objective

The research in this dissertation aims to improve the early stage design process of service vessels by reducing the effort required to investigate thoroughly a large number of alternatives (with variations in system choices, configuration and performance levels) during the early stage design of service vessels.

Hence, the research objective becomes:

- **Reduce the effort required to generate and investigate a large number of alternative ship designs during the early stage design of service vessels**.

Chapter 2 starts the development of the new approach by reviewing existing approaches that offer computer-based support for the early stage design of ships. Chapter 3 then develops the outline of the novel approach, before Chapter 4, 5 and 7 develop it in full. Chapter 6 applies the approach before Chapter 8 uses the results to conclude this dissertation by reviewing whether the research objective is met.

Chapter 2

A review of computer-based ship design methods from literature

2.1 Introduction

Chapter 1 introduced service vessels and highlighted the considerable effort it takes for a naval architect to thoroughly investigate a sufficient number of interesting combinations of systems, configurations and performance levels in the limited time available during early stage design. Section 1.5 argued that the consequence is that risk will increase and the ability to improve competitiveness will reduce. It is this consequence that warrants the development of a new approach.

For this reason, the purpose of the research presented in this dissertation (Section 1.7) became the development of an approach that enables the investigation of a large number of alternatives with different system choices, configurations and performance levels during the early stage design of service vessels.

The development of this approach obviously builds upon previous research in the field of early stage ship design. Therefore, this Chapter will present a brief literature review with a focus on computer-based support for early stage ship design. Objective of the literature review is twofold.

- First, to investigate whether existing approaches can support the investigation of a large number of alternatives during the early stage design of service vessels.
- Second, to investigate whether modifications to existing approaches can support the investigation of a large number of alternatives during the early stage design of service vessels.

Starting point for the literature review is Section 2.2, which discusses different methods to describe a ship during the early phases of the design process. Since the early stage ship design process revolves around developing such descriptions, their introduction is an essential requisite for the literature review itself, which is presented in 2.3.

2.2 Ways to describe a ship

Designing a ship means, among other things, describing her. Ultimately, the goal of designing is to create a description of the ship at a level of detail suitable for production (see *Van der Nat [120]*) and, hence, the design process focusses on developing such a description.

Building a ship is, however, a major step. Before committing to it, one uses the earlier design phases to build up confidence in the ship's design. This involves creating numerous descriptions (at a gradually increasing level of detail) and subjecting them to performance predictions, for instance, to see if they comply with non-negotiable design requirements.

This dissertation uses the three types of descriptions proposed by *Van der Nat [120]*; each is discussed below.

- Numerics. A numerical description of the design concerns aspects such as the ship's size, displacement, speed or wave-induced accelerations. A numerical description is determined by mathematical relations made up by numerical parameters. Note that, mathematical relations can become rather complex, e.g., notably those used by advanced prediction tools (such as in *Hoekstra [59]*).

- Geometry. Geometry concerns the description of the spatial aspects of the ship, i.e., both shape of the ship's systems and their position in the configuration. Common examples include the shape of the hull and superstructure as well as the shape and positions of systems (attributing different positions for systems was one of the options discussed in Section 1.3). Geometry can be described using a variety of different computational methods such as NURBS[1] (see *Piegl and Tiller [99]*), Constructive Solid Geometry (see *Dykstra and Muuss [40]*) or subdivision surfaces (see *DeRose et al. [37]*).

- Topology. Topology in this dissertation deals with the connectivity between parts of the ship (one of the system options discussed in Section 1.3). Common examples include the layout of a chilled water distribution system or the connectivity between the different systems in the propulsion plant. Note that *Van der Nat [120]* used topology to describe the relative positions of parts of the ship, which is treated as geometry in this dissertation.

Section 1.7 stated the need to be able to investigate many alternatives with different system choices, configurations and performance levels during early stage ship design. This obviously influences the types of descriptions to be used.

For example, system choices can be expressed using numerical parameters, while describing a system's shape obviously needs geometry. Describing a configuration needs geometry as well, while a prediction of post-hit weapon availability on a warship requires a description of the topology of the ship's services networks used by the weapon systems, as well as numerical description to express their performance, e.g., stating the available and required cooling power.

As a result, the approach to be developed in Chapter 3 must use all three types of descriptions concurrently: numerical, geometrical and topological.

[1]NURBS stands for Non-uniform rational B-spline

2.3 Prior efforts in computer-assisted ship design

Computing power increased steadily for the past decades and computer-based support for early stage ship design has been employed and developed since the sixties. As time progressed, computing power increased resulting in new possibilities that altered the capability of computer support in ship design. A brief overview is presented below.

First to develop were concept exploration models, e.g., as discussed in *Eames and Drummond [41]* and *Nethercote and Schmitke [86]*. They consist of a predominantly numerical design description (Section 2.2) that, combined with a collection of simple ('regression-based') prediction tools, enables a rapid investigation of various alternatives. Concept exploration models have several benefits. First, changes to the numerical description are straightforward and can be carried out by systematic variation. Second, combining multiple prediction tools helps to create a set of alternatives suitable to investigate trade-offs between negotiable design requirements. *Wolff [139]*, *Nixon [91]* and *Lamerton et al. [71]* present three more recent applications.

The next development combined numerical concept exploration models with mathematical search algorithms (commonly called optimisation algorithms). *Mandel and Leopold [77]*, *Keane et al. [63]*, *Neu et al. [87]*, *Brown and Salcedo [20]* and *Stepanchick and Brown [111]* provide examples of such applications. Search algorithms employ numerical **ratings**, i.e., objectives and constraints, to identify 'promising' designs to pursue; their development enabled the abandonment of systematic variation in favour of a more focussed and more efficient search. Their development resulted from the availability of suitable search algorithms and sufficient computational power to use them, while the numerical design description (discussed above) naturally lends itself for automated variation. Practical benefit for ship design purposes is the ability to search for designs meeting a pre-defined set of design requirements (which Section 1.4 discussed).

Further increases in computational power enabled computer-based visualisation and alteration of the ship's geometrical and topological description, in addition to the numerical description already used. This first replaced drawing board and paper directly, but gradually more advanced capabilities were added, e.g., defining a ship in three dimensions. Changes to the geometry description were manual at first, but soon user-defined mathematical relations defining position and shape enabled the parameterisation of geometry and topology, allowing them to be changed solely by altering parameter values. Parametric capabilities expanded ultimately to geometric constraint-solving, e.g., see *Harries [56]* for an application in hull form design. *Andrews and Pawling [8]* and *Van Oers et al. [131]* both used parametric geometry descriptions in early stage ship design.

A parametric geometry and topology description enables the use of techniques already employed for numerical concept exploration models, i.e., systematic variation and search algorithms, to generate many alternatives with the purpose of identifying promising designs. Examples include hull shape variation (*Harries [56]*), internal arrangement (*Lee et al. [72]* and *Nick [88]*), configuration of equipment aboard an offshore platform (*Smith et al. [110]*), routing and topology variation of piping and ducting (*Asmara and Nienhuis [12]*), structural arrangement (*Rigo [105]*) and watertight subdivision (*Boulougouris and Papanikolaou [18]*).

Parametric geometry descriptions also enabled more advanced prediction tools to be used; the most advanced tools solve a set of discretised physics-based equations (e.g., the Navier-Stokes equations in *Hoekstra [59]*) on a spatial grid distributed over (part of) the geometric ship description. These prediction tools broaden the range of designs for which performances can be predicted accurately. Recently, several applications combined such tools with parametric geometry and search algorithms, e.g., as in *Peri and Campana [97]*, *Maisonneuve et al. [75]*, *Valdenazzi et al. [117]*, *Giassi et al. [46]* and *Kulkarni et al. [70]*.

The increased ability to create, process, store and retrieve information lead to a massive increase in both the level of detail and the comprehensiveness of design models. This has gradually built up the information content of such integral design models up to the point where complete virtual descriptions of the ship are available. Typically, such product-data models are used during the engineering stages of the design process. Conceptual design applications are discussed in *Andrews [3]*, *Van der Nat [120]*, *Andrews and Pawling [8]*, *Andrews and Pawling [9]* and *Van Oers et al. [131]*. The increased level of detail and comprehensiveness had an important consequence: design changes required considerable human interaction.

The ability to generate a large number of alternatives enables naval architects to identify important trends in the set of designs. It simultaneously introduces a problem, as only a few designs are chosen for detailing in the later design stages. Therefore, a suitable design needs to be selected from the multitude of alternatives. Supporting this effort are post-processing tools. Such tools provide the naval architect with the ability to filter information and, most importantly, identify trends using a graphical user-interface. See *Sen and Yang [108]*, *Stump et al. [112]*, *Eddy and Lewis [42]*, *Maisonneuve et al. [75]* and *Valliyappan and Simpson [118]*. *Stump et al. [112]* is of particular interest, as their approach allows one to vary one's preferences (by altering user-defined weights) during the investigation to study the impact on the desirability of particular designs. Note that post-processing tools do not make decisions. Humans make decisions based on the information made accessible by post-processing tools.

Closure

From this overview, one may conclude existing approaches offer sufficient support to rapidly generate and investigate a large number of different ship configurations using a combined numerical-geometrical and topological description.

A more nuanced conclusion is warranted, however, and Sections 2.4 to 2.6 discuss some general observations, beneficial trends and problems, respectively. Section 2.7 draws conclusions from them to answer the questions laid out in Section 2.1.

2.4 General observations

Four general observations on the references from Section 2.3 are presented first. They concern the iterative design process (Section 2.4.1), inventiveness (Section 2.4.2), human decision-making (Section 2.4.3) and, lastly, the role the naval architect has (Section 2.4.4).

The observations in Sections 2.4.1 to 2.4.4 are intended to be neutral. As such, stating that a search algorithm can make decisions using pre-defined rules does not mean one should apply it immediately. Instead, it merely indicates that it is possible to do so if deemed appropriate. Therefore, the general observations will be used to formulate the novel design approach in Chapter 3, but only after careful consideration whether and when their application is appropriate.

2.4.1 Iterative design process

All references used an iterative process more or less similar to that shown in Fig. 2.1. The first step involves inventing a ship description (a task performed by humans, as discussed in Section 2.4.2). Subsequently, the performances, e.g., intact stability, are predicted and analysed against the design requirements (discussed in Section 1.4). Next, humans or computer software have to decide upon design changes aimed to make performances meet the requirements. These changes have to be implemented by altering the description, completing the process. The process continues until a description is available that meets the requirements.

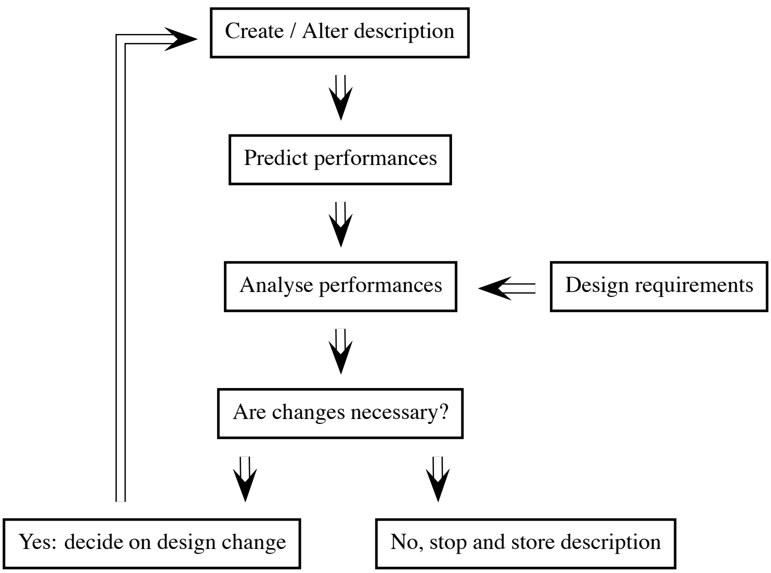


Fig. 2.1 – Simple design loop

Numerous applications in Section 2.3 combine several design loops from Fig. 2.1 in a single comprehensive integral design model, capable of describing and altering the design and predicting multiple performances (e.g., *Van der Nat [120]*, *Andrews and Pawling [8]* and *Nixon [91]*).

The iterative nature of the process in Fig. 2.1 is caused by uncertainty. One creates an initial description based on assumptions and opinions, without certainty that it will meet the requirements. If the design does not meet the design requirements, one again devises design changes without certainty they will suffice. Fortunately, performing a sufficient number of iterations helps to investigate different design changes and, crucially, helps the naval architect to learn what changes to the description will ensure compliance with the design requirements.

Most importantly, the iterations in Fig. 2.1 need to be performed for each and every alternative of interest (Section 1.3) until compliance with non-negotiable design requirements is ensured. This partly explains the large effort required to investigate the many alternatives, as outlined in Section 1.5 (Section 2.4.2 discusses the other cause).

Note that the simple process in Fig. 2.1 does not distinguish between the negotiability of design requirements, nor does it consider any priorities that requirements may be attributed with. Instead, the process merely uses generic design requirements to explain their role in the iterative process: a measure to compare a design's performances against. As such, Fig. 2.1 does not distinguish between 'sizing', 'balancing' and 'performance prediction' as *Van der Nat [120]* does in Fig. 4.10.

Both humans and computer software (e.g., a search algorithm) can perform design iterations once the initial description has been invented, though both use different ways to come up with design changes. Humans can invent design changes during the iterative process, while software can use rule-based changes invented beforehand by humans (at least for the intent and purpose of the research in this dissertation). The use of such pre-defined rules means computer software can perform design iterations considerably faster than humans, a feature which Chapter 3 will exploit to address the problems outlined in Chapter 1.

Software's use of rule-based changes may appear to be very restrictive. This need not be the case as even a small set of simple rules can result in a massive number of possible alternatives. Whether such a large number of possible alternatives occurs depends on the particular rules. Still, the author deems computer software to be unable to venture outside the range of possibilities created by human inventions, even when the range of possibilities is very large.

2.4.2 Inventiveness

All references discussed in Section 2.3 relied heavily on human inventions and hence, human inventiveness. Inventiveness is considered to be human creativity applied to solve practical problems, unlike the arts, for instance.

More specifically, before the naval architect can apply computer support during early stage ship design, he first has to invent, for example, the initial ship description, performance prediction tools, design requirements, and most importantly, the design changes used to improve performances. Once invented, software (such as a search algorithm) can help to generate suitable designs, but -most importantly- these designs can only fall within the range of possibilities that are created by the naval architect's inventions.

These inventions required for the application of computer-based support form a considerable investment of human effort and should, therefore, be reusable for multiple ship types and projects. Moreover, the human effort required to develop the ship description is affected proportionally by its level of detail. Therefore, more detailed descriptions require more 'details' to be invented and as such, require more human effort. The latter is amplified proportionally when considering multiple alternatives, causing the problems outlined in Section 1.5.

Furthermore, inventions need not be revolutionary. They may concern very simple things that must be adjusted to be able to perform the design iterations (e.g., converting a particular computer file from one format to another). In all cases, inventions require a dedicated human act.

The naval architect's role in invention has two important consequences for the application of search algorithms (and other means of using computer software to automate the generation of design changes) used by references to perform iterations.

- First, the search algorithm itself has to be invented, as do the parameters, objectives and constraints it uses, as well as the translation of parameters into a ship description.

Hence, one must add the time needed for such inventions (made in preparation for the application of search algorithms) to the time required to perform the search process itself, in order to prevent chasing 'false' economies.

- Second, the role of humans in invention also affects the degree with which the speed - and hence number- of design iterations can be increased. The reason is that this increase can only be realised when human inventions (which are time consuming) are avoided when performing design iterations, in order to prevent the human from becoming the bottleneck.

Chapter 3 considers the impact of human inventions on the application of search algorithms in detail.

2.4.3 Human decision-making

Another topic shared by all references in Section 2.3 is the intention to support human decision-making. As discussed below, decision-making in the context of the early stage design of service vessels mainly concerns deciding upon the suitability of particular alternative based on, among other things, the design requirements. Other kinds of decisions are also important, e.g., the prediction tools to be used, or the appropriate level of detail of a ship description, but these are not discussed in this Section for reasons of brevity.

Decisions are considered to be human inventions (discussed in Section 2.4.2). Therefore, computer programs, such as a search algorithm, can only decide upon the suitability of an alternative according to pre-defined rules invented earlier (Section 2.4.1).

Two types of decisions are considered by the references in Section 2.3.

- Threshold decisions. Threshold decisions typically concern a comparison against a non-negotiable design requirement (Section 1.4). If a particular alternative exceeds a threshold (or remains below a threshold), one can decide that it is acceptable. A common example of a threshold decision is to regard a design as acceptable if it exceeds a required minimum meta-centric height to ensure sufficient initial stability. Typically, there is widespread agreement on the formulation of these thresholds, e.g., warship stability requirements as proposed by *Sarchin and Goldberg [107]*.

- Trade-off decisions. When taking a trade-off decision, the naval architect finds a compromise between a negotiable design requirement and one or more other design requirements (Section 1.4). For example, one can increase the ship's beam to improve stability at the cost of an increase in resistance. Trade-off decisions are based on priorities, i.e., opinions that deem one aspect more important than another.

  The reliance on human opinions is of particular interest, for two reasons. First, opinions vary among people, which makes it difficult to get agreement on a particular trade-off. Second, and most importantly, human opinions can change on the basis of new information, e.g., creating a ship design may reveal issues not considered previously. As a result, opinions may need adjustment due to insight gained in prior design iterations (hence, the 'wicked problem' discussed in Section 1.5). In particular, studying the ship description (the outcome of the design process most familiar to the naval architect) is especially useful to see if trade-offs need adjustment.

This need for adjustment is a crucial characteristic of trade-off decisions: they are considered to be **implicit** at the start of the design process. It is difficult, therefore, to establish a-priori what a suitable compromise will look like. Hence, a trade-off decision can be finalised only once all relevant consequences have become known, i.e., once humans have gained all insight needed (resulting from the creation of a ship design based on the trade-off decision) to determine whether changes to the decision are actually necessary.

This implicit nature of trade-off decisions will play an important role in the development of the new approach in Chapter 3.

The references from Section 2.3 support the naval architect in taking both types of decisions in two different ways.

- A-priori. One can formulate rules stating the required trade-off and threshold decisions before performing design iterations. Once defined, computer software can use them during iterations to guide the design process.

  The a-priori approach does enable one to use search algorithms to ensure compliance with a pre-defined set of requirements, e.g., they can ensure the ship floats. Still, in such cases, the need to pre-define decision rules employed by search algorithms, e.g., as by *Lee et al. [72]* and *Nick [88]*, means the ability to adjust trade-off decisions to cope with their implicit nature is limited (see discussion above).

- A-posteriori. Alternatively, one can avoid pre-defining decision rules. Instead, one can generate numerous alternatives and compare them to provide the naval architect with the insight needed to take both threshold decisions and trade-off decisions. This kind of support was used, for instance, by *Maisonneuve et al. [75]*, to illustrate trade-offs between payload, sea-keeping and resistance of a fast ferry.

Several references combined a-priori and a-posteriori support, e.g., *Maisonneuve et al. [75]* used an a-priori threshold decision to constrain displacement, while using a-posteriori decisions to trade-off resistance against sea-keeping for a fast ferry. Chapter 3 will consider support for both threshold and trade-off decisions in more detail.

Lastly, having humans make decisions a-posteriori could help instil a sense of acceptance for the resulting designs (Section 1.7). This is important because ultimately, humans decide which alternative is chosen for detailing in the later stages of the design process. Simultaneously, using human decision-making in an a-posteriori manner could increase the time required to perform design iterations, as humans will -by definition- need to become more involved. This means one may encounter a trade-off between speed of iteration and achieving acceptance for the results. Chapter 3 will discuss the generation of acceptance and its related issues in more detail.

2.4.4 The naval architect's role

The discussion on inventiveness and decision-making in Sections 2.4.2 and 2.4.3 illustrates how the naval architect's role in computer-assisted ship design is far from diminished. Instead, when properly supported, this role is strengthened, enabling the naval architect to focus on inventing solutions and taking decisions.

Still, the way the naval architect uses computer support requires proper attention to ensure both human and computer software see their strengths magnified. Chapter 3 discusses the distribution of work between human and software in detail.

2.5 Beneficial trends

Section 2.4 presented several general observations related to all references in Section 2.3.

This Section presents specific trends that can help the naval architect investigate numerous alternatives during early stage design of service vessels (the research objective stated in Section 1.7). How each particular benefit helps the naval architect is discussed below.

1. The number of designs under consideration increased as more computational power became available. This trend occurred for both numerical descriptions and geometrical and topological descriptions, e.g., *Stepanchick and Brown [111]*, *Lee et al. [72]* and *Maisonneuve et al. [75].*

   Still, references who used more detailed and more comprehensive geometrical descriptions (e.g., *Van der Nat [120]*, *Andrews and Pawling [9]*) were able to create fewer alternatives (typically a dozen or less) than those references who used less detailed and less comprehensive geometrical descriptions (e.g. *Harries [56]*).

   Nonetheless, the references illustrate that the most important need can be satisfied: the ability to generate numerous alternatives sufficiently fast to be useful during early stage design .

2. The description of the ship shifted from predominantly numerical models to increasingly detailed geometrical, topological and numerical descriptions while the level of detail also increased. This trend is crucial, as the design of service vessels warrants the generation of configuration from the earliest moment on (Section 1.2), which is supported by the ability to generate and alter geometrical, topological and numerical descriptions at a level of detail deemed suitable by the naval architect.

3. The development of parametric numerical and geometrical descriptions enables all features of a ship configuration, i.e., system type, number, connectivity and their shape and size, as well as their position, to be changed solely by altering the value of numerical parameters. This is a requisite to exploit the ability of computer software, e.g., search algorithms, to perform design iterations with increased speed by avoiding reliance on human interaction during these iterations.

4. The development of search algorithms, combined with availability of parametric geometry and topology descriptions, enabled the search for designs with user-defined properties, e.g., to focus the design effort towards designs meeting non-negotiable design requirements. This search ability was complemented with post-processing tools to investigate the resulting set of designs.

5. The nature of design support software becomes increasingly integrated and combines both numerical and geometrical descriptions together with several performance prediction tools. This trend enables the early assessment of compliance with the non-negotiable design requirements and, equally important, the identification of necessary trade-offs

between negotiable design requirements and thus whether performance levels are sufficient. For example, the impact of system choices was considered by *Van der Nat [120]* and *Andrews and Pawling [9]*, the merits of different configurations were considered by *Andrews and Pawling [8]* while different performance levels were studied by *Maisonneuve et al. [75].*

The five trends illustrate that the potential exists to develop computer-based support for the thorough investigation of numerous alternatives. Successful exploitation of the trends should enable the naval architect to study and identify interesting combinations of systems, configurations and performance levels early on. In turn, this should help improve the competitiveness of service vessels.

Still, successful exploitation is not necessarily straightforward, as Section 2.6 will reveal. It identifies and discusses two problems that hamper realising the benefits from the five trends outlined above.

2.6 Problems

Two problems were observed in the references from Section 2.3. Both problems make it difficult to reap the benefits from the trends discussed in Section 2.5. They are discussed below.

- **The ability to handle large design changes without human interaction**. The first problem concerns the magnitude of the design changes in relation to both the type of description used and the need to use human interaction to make the changes.

  In early stage ship design, one typically see large changes made to the initial description, changes necessary to cover the many options of interest (Section 1.3). Therefore, computer support must be able to cope with such changes at a level of detail that provides the required accuracy (*Van der Nat [120]* and *Andrews [5]*). The ability of the references to cope with such changes differed markedly and two categories were identified.

  - First, large changes are possible when numerical description was used, e.g., as in *Eames and Drummond [41]* or *Nixon [91].* In these cases, the large changes could be generated by both by human interaction and by a search algorithm (the latter using pre-defined rules).
  - Second, large changes could only be achieved when a geometrical description was used by relying on extensive human interaction, e.g., as by *Van der Nat [120]*, *Andrews and Pawling [8]* and *Van Oers et al. [131].*

    Several applications combined search algorithms with parametric geometry to perform design iterations without human interaction (e.g., *Harries [56]*, *Lee et al. [72]* and *Rigo [105]*). Unfortunately, these applications only allowed minor changes to a single part of the ship's design, e.g., by changing bulbous bow shape (as in *Valdenazzi et al. [117]*).

  This observation gives rise to the first problem: the apparent inability of search algorithms (and other ways to perform design iterations automatically) to instigate large design changes to a geometric description of the ship. This is problematic, as it means that the application of search algorithms can only handle minor changes, changes that too small to cover the many options facing the naval architect (Section 1.3).

- **The ability to handle concurrent changes to the entire ship description without human interaction**. The second trend is related to the first. The changes during early stage design are not only large (as discussed above), but obviously affect all parts of the ship description, e.g., hull, superstructure, decks, bulkheads and systems. Therefore, computer support must be able to cope with both large and concurrent changes to different parts of the ship's description to be of any use in early stage design.

  Again, the references discussed in Section 2.3 can be categorised according to the concurrency of changes, e.g., whether they allow both hull shape and internal arrangement to change. The distinction is made between many or few concurrent changes.

  - **Many concurrent changes**. Many concurrent changes occurred in the same two situations as with the occurrence large design changes. Either a numerical description was used, possibly in combination with automated design changes, e.g., as in *Eames and Drummond [41]* or *Brown and Salcedo [20]*, or a detailed geometrical description was used in combination with extensive human interaction, e.g., as in *Van der Nat [120]*, *Andrews and Pawling [8]* or *Van Oers et al. [131]*.
  - **No or few concurrent changes**. Changes to a single part of the ship occurred in those references, e.g., hull shape (e.g., *Harries [56]*), that combined a geometrical description with search algorithms. Minor exception was *Giassi et al. [46]*, who combined variation of the shape of a fin stabiliser with that of the structural arrangement.

  The differences reveal the second problem: the apparent inability of search algorithms (and other ways to perform design iterations automatically) to handle both large and concurrent changes to the entire geometrical ship description. Handling large and concurrent changes appear to require human interaction.

  This is problematic as the application of search algorithms to rapidly perform design iterations is useless when only minor changes to a single part of the design can be considered. The ability to support large and concurrent changes to the entire ship description by changing parameter values is crucial to enable the investigation of numerous alternatives during the early stage design of service vessels.

Both problems can be generalised, as shown in Fig. 2.2. It visualises the relation between the type of description (and the amount of detail) and, respectively, the magnitude of design changes, the concurrency of design changes, and the number of designs that was generated, for three references considered representative. First, *Peri and Campana [97]* varied frigate hull geometry using a search algorithm to improve hydrodynamic performances; second, *Andrews and Pawling [8]* used a human-centered geometrical approach for the early stage design of a wide range of vessels. Third, *Stepanchick and Brown [111]* used a predominantly numerical design description, in combination with a search algorithm, to investigate a range of system choices and performance requirements for a destroyer-type warship.

Fig. 2.2 visualises the two problems identified earlier. The required capability in each of the three images in Fig. 2.2 is indicated by the gray triangle located in the upper-right corner. Ideally, the required capability of the approach combines a geometrical description of the complete ship, with the ability to undergo major changes affecting the entire description, while being sufficiently fast to consider hundreds or thousands of designs.

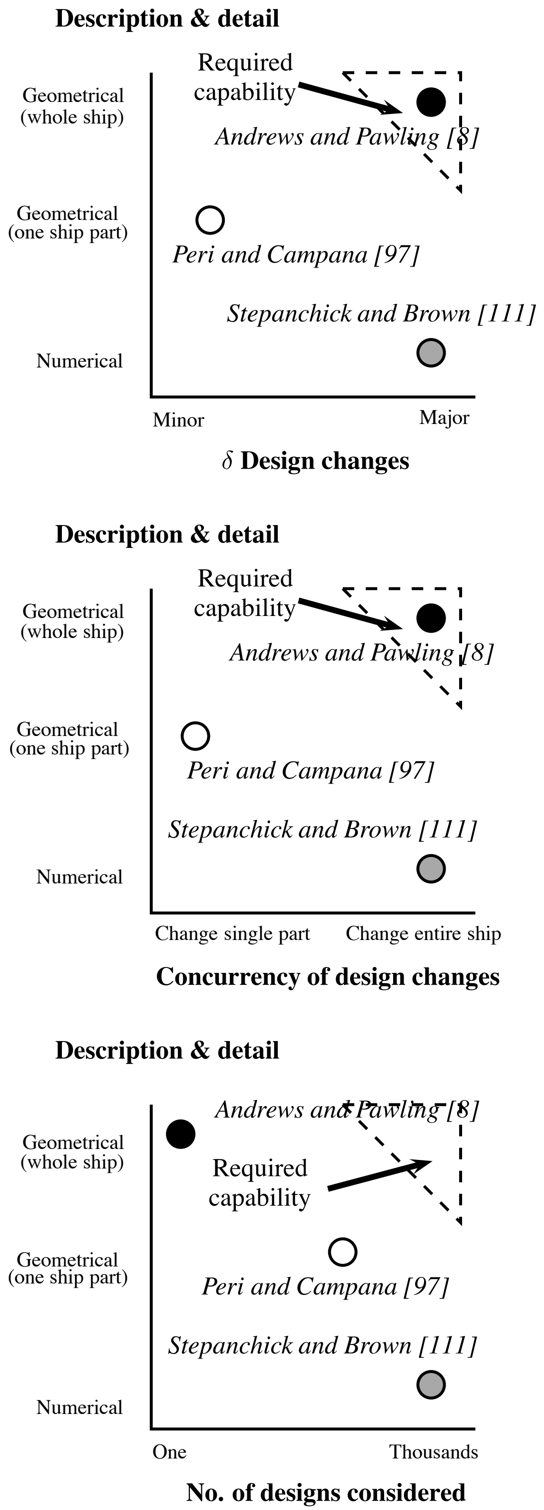


Fig. 2.2 – Qualitative representation of problematic trends

Unfortunately, Fig. 2.2 also shows that none of the three references has the required capability. *Andrews and Pawling [8]* uses a geometric model in combination with considerable human interaction, which allows major changes to the entire ship. However, the reliance on human interaction simultaneously limits the number of designs *Andrews and Pawling [8]*'s approach can consider. *Stepanchick and Brown [111]* considered large changes to the entire ship design for thousands of alternatives, but only used a numerical description at a low level of detail, while *Peri and Campana [97]* lies somewhere in-between *Andrews and Pawling [8]* and *Stepanchick and Brown [111]*.

This means the references have, at most, two of the three capabilities necessary to resolve the problems outlined in Chapter 1, with the closest one being *Andrews and Pawling [8]*. Therefore, the references in Section 2.3 do not appear to solve the problems from Section 1.5 outright. At best, they can serve as a starting point to be improved upon.

Still, the question is whether the two problems result from distinct choices made by the authors of the references discussed in Section 2.3, or whether they are truly unavoidable? Fortunately, one reference provides a glimpse of the answer. *Valdenazzi et al. [117]* varied bulbous bow shape to reduce wave resistance, but simultaneously required other hull changes to be minor, e.g., required displacement had to remain within a one percentage band of the starting point.

In this particular reference, the constrained nature was purposely chosen according to *Valdenazzi et al. [117]*, because: '*at the stage when the shape of the bulb is closely studied, the main ship particulars and internal arrangements have been already decided upon*'.

Such a constrained nature simplifies the problem considerably and enables a focus on the problem at hand: finding a proper bulb shape, at the cost of not resolving the larger problem with potentially larger improvements: improve hull shape, bulb shape and arrangement concurrently.

The fact the limited variations were purposely chosen -for this reference, at least- indicates it should be possible to address the two problems with the application of search algorithms that were identified. Chapter 3 will explain the approach used to resolve them.

Section 2.7 concludes this Chapter by summarising the main findings from the literature review and using the results to answer the questions laid out in Section 2.1.

2.7 Conclusions: the ability of existing approaches to address the problems identified in Chapter 1

This Chapter reviewed existing computer-based efforts that support early stage ship design. It established beneficial trends, problems, as well as general features shared by the references.

What remains is a reflection upon the two questions posed in Section 2.1 which started the literature review. This Section answers them using the literature review from Section 2.3 and the subsequent insight gained in the discussions in Section 2.4 to 2.6. The two questions and their answers are discussed below.

- **Whether existing approaches can support the investigation of a large number of alternatives outright?** The answer to the first question is perhaps the most important: whether existing approaches can support the investigation of a large number of alternatives outright, i.e., without modifications. This establishes whether there is a need for a novel approach at all. There is, as the reasons discussed in Section 2.6 illustrate.

Section 1.2 argued the need to consider the ship configuration from the earliest moment on (which requires geometrical descriptions), while Section 1.5 discussed the need to create many alternatives with different system choices, configurations and performance levels, to thoroughly investigate the many options facing the naval architect.

Unfortunately, the references from Section 2.3 are unable to combine the ability to investigate many alternatives with the use of a geometrical description that describes the entire ship, while also being able to undergo both large and concurrent changes to the entire ship description (this was visualised in Fig. 2.2). The reason is that handling large and concurrent changes currently requires extensive human interaction, which appears to prevent the use of search algorithms to investigate a large number of alternative ship designs.

Therefore, the development of a new approach is warranted (thus confirming the conclusions in Section 1.7) to reduce the human effort required for the thorough investigation of the many alternatives of interest. Its development can benefit considerably from the existing approaches, as is seen from the answer to the second question.

- **Whether modifications to existing approaches can support the investigation of a large number of alternatives?** Since existing approaches do not suffice, the next best thing is to reuse beneficial parts of existing approaches to reduce human effort and increase the number of alternatives that can be investigated.

  The beneficial trends outlined in Section 2.5 serve as a starting point, with a special focus on three trends in particular. Firstly, the ability of search algorithms to search for designs meeting a set of requirements. Secondly, the ability to use changes in parameter values to change all parts of a ship description (though not concurrently, and with generally minor changes) and thirdly, the ability to investigate different trade-offs by generating numerous alternatives using integrated design models that combine a ship description with several performance prediction tools. Chapter 3 discusses the required modifications in more detail.

Closure

In summary, no existing approach was found that, without modification, enables one to address the problem outlined in Chapter 1: the prohibitive human effort required to thoroughly investigate the many alternatives of interest.

Therefore, new design approach is required, which is to be developed in Chapter 3. Useful parts of the existing approaches discussed in Section 2.3 will serve as a starting point. This new approach does not merely combine existing approaches from the references in Section 2.3. Instead, the approach will be built from the ground up, using novel methods and by using elements from earlier references. This helps to identify issues that arise when combining different elements of previous work and also prevents ‘adjusting the problem to the solution’.

Chapter 3

Developing the novel approach

3.1 Introduction

Chapter 1 discussed the need to thoroughly investigate the many options a naval architect faces during the early stage design of service vessels. This investigation, in turn, requires the generation of a large number of alternative ship designs.

Unfortunately, the literature review in Chapter 2 illustrated that existing design approaches do not support the creation of so many alternatives (with a geometric ship description) without requiring a prohibitive human effort. As a result, Chapter 2 concluded that the development of a novel design approach is warranted and listed trends this novel approach may benefit from.

This Chapter develops the outline of this new approach. The approach aims to generate many 'promising' alternatives using a parametric geometrical ship description integrated with a search algorithm. With the resulting abundance of alternatives, the naval architect can investigate the many options of interest (Section 1.3). The hope is that the application of search algorithms will reduce the human effort to manageable proportions, something Section 3.7 will reflect upon to establish whether a reduction of human effort will indeed be attainable.

The use of search algorithms has considerable impact on the design process. Most importantly, the correct application of search algorithms to solve practical ship design problems is anything but straightforward (as discussed in *Andrews [4]*). Therefore, this Chapter considers how to best use search algorithms to support the investigation of a large number of alternatives during the early stage design of service vessels. To do so, it builds upon the references, observations and trends from Chapter 2. Several other references dealing with specific features of search algorithms are also used.

The Chapter starts in Section 3.2 by questioning why the use of search algorithms should be able to reduce the human effort required to investigate many alternative ship configurations.

3.2 Why use search algorithms?

Section 2.4 introduced the simple iterative design loop (shown in Fig. 2.1) and explained how both humans and computer software can use it to perform design iterations. This Section discusses two perceived benefits observed in Section 2.3 that follow from using search algorithms to perform design iterations.

- First, search algorithms can increase the number of iterations performed in a fixed amount of time in comparison to a naval architect. This increase is possible thanks to the exploitation of computational power combined with a lack of direct human interaction during the design iterations.

- Second, search algorithms can search for designs that meet pre-defined ratings, which enables a focussed and efficient generation of ship designs that meet design requirements for which such ratings are formulated. This relieves the naval architect from the effort required to ensure compliance with these design requirements.

The two benefits provide the starting assumption for the work presented in this dissertation: that search algorithms can help to address the problems laid out in Chapter 1.

Realising these benefits requires addressing several potential drawbacks of search algorithms, which Section 3.4 and 3.5 respectively introduce and address. Before doing so, a more in-depth introduction to search algorithms is required, which Section 3.3 provides.

3.3 Structure of a search algorithm

This Section discusses the structure of a generic search algorithm. This forms the basis for the development of the novel approach outlined in Section 3.6, which will generate the large number of alternatives. The starting point for the discussion is Fig. 3.1 (which shows part of a design loop similar to the one discussed in Section 2.4.1).

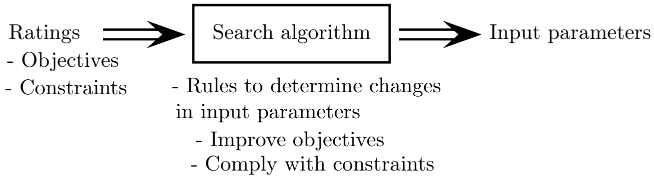


Fig. 3.1 – A search algorithm with ratings and input parameters

Numerical parameters

The application of a search algorithm revolves around the use of numerical parameters (as shown in Fig. 3.1). There are two types of parameters: input parameters and ratings. Both are discussed below.

- **Input parameters**. Fig. 3.1 shows the search algorithm's **output** consists of input parameters. They are called input parameters because they form the input for the parametric ship description, which will be discussed below. These parameters are invented by humans, who also determine their number, their type (continuous, discrete or combinatorial), as well as the range of numerical values they may take.

  The search algorithm uses input parameters to instigate changes to the ship description by altering their values, e.g., vary the parameter determining the ship's length to change it from 100 to 150 m.

  The number of parameters determines the number of parts of the ship that can undergo changes independently. Investigating the impact of the many options independently (see Section 1.3) therefore requires a large number of input parameters.

- **Ratings**. The search algorithm uses numerical ratings, i.e., objectives and constraints, as **input** to guide the search process. Humans invent ratings, which express the characteristics of interest for the search process in numerical values. For example, a search algorithm can use installed propulsion power as a rating to search for ship designs with reduced propulsive power. The number of characteristics of a design that a search algorithm can consider independently depends on the number of ratings used, similar to input parameters. For example, searching for both improved resistance and sea-keeping independently requires two ratings.

Converting ratings into input parameters

Fig. 3.1 shows that a search algorithm uses prior ratings to determine new, altered values for the input parameters. This process forms the core of the search algorithm and determines, among other things, the complexity of search problems it can tackle. The conversion process uses predefined, mathematical rules; rules invented by humans. The exact rules differ per type of search algorithm, as well as the type of input parameters and the number of ratings used. These differences in rules give rise to a multitude of search algorithms, ranging from simple algorithms, e.g., *Nelder and Mead [85]*, to complex algorithms, e.g., *Deb et al. [32]*. Each search algorithm is tailored to tackle particular categories of search problems (see *Hillier and Lieberman [58]* for an overview).

The rules used to translate ratings into changed values of input parameters commonly distinguish between two kinds of ratings: objectives and constraints. Both are discussed below.

- **Objectives**. Objectives concern ratings that express characteristics of the ship that should be improved during the search effort, i.e., minimised or maximised. Examples include of cost and resistance. For minimisation, designs with lower ratings are considered to be more 'promising' than those with higher ratings. Common feature of objectives is the lack of a threshold (discussed in Section 2.4), i.e., objectives are improved until no further enhancement is possible. Objectives are suited to investigate the impact of negotiable design requirements (discussed in Section 1.4).

- **Constraints**. Constraints concern ratings that express characteristics that should be safeguarded during the search process, i.e., their numerical value should lay within user-defined bounds or comply with user-defined thresholds (see Section 2.4). Examples are the achievement of sufficient buoyancy and stability. Constraints are suited to safeguard compliance with non-negotiable design requirements (discussed in Section 1.4).

Converting input parameters into ratings

Something is missing from Fig. 3.1: the conversion of input parameters into ratings (objectives and constraints). The reason is that, so far, the discussion of search algorithms has been generic. It did not consider **what** the input parameters change, only **how** they change it. More specifically, the search algorithm itself has no knowledge of the meaning humans attribute to both its input parameters and ratings. It merely uses its own rules to alter input parameter values based on the values of constraints and objectives.

Fortunately, this means the same input parameters in Fig. 3.1 can be used to vary aircraft configurations (e.g., *Buonanno and Mavris [22]*), vehicles (e.g., *Miao et al. [82]*) and ships (e.g., *Nick [88]*). To generate a set of alternative service vessels, one obviously uses input parameters to vary the design of such ships during the conversion of input parameters into objectives and constraints.

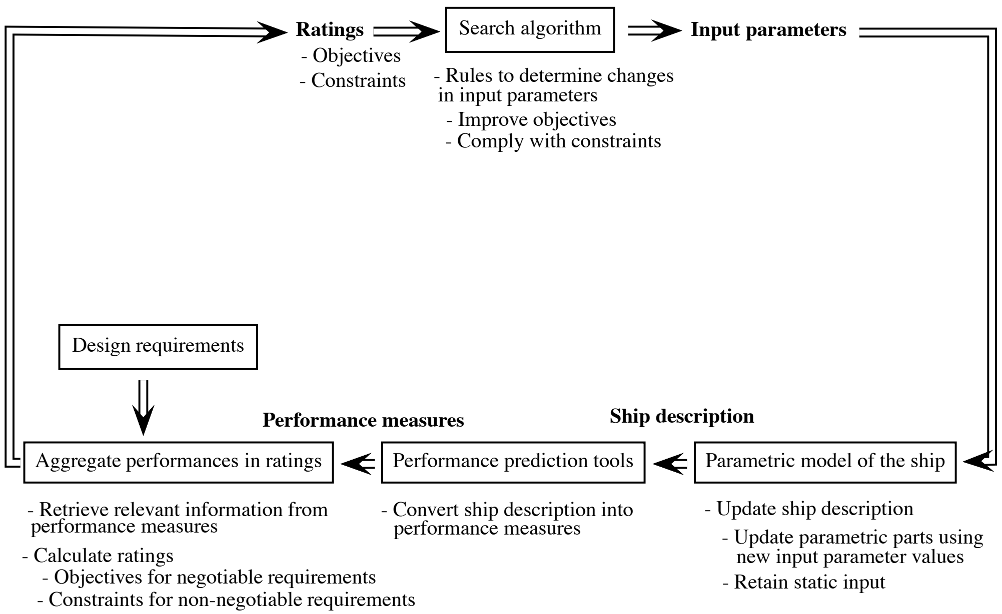


Fig. 3.2 – Fig. 3.1 updated with a parametric model of the ship, performance prediction tools and calculation of ratings

This conversion of input parameters into objectives and constraints takes place in three steps, as is shown in Fig. 3.2. They are discussed below.

- **Parametric model**. The first step uses the input parameters to alter the description of the ship. This requires a parametric model that translates changes in input parameter values into changes to the ship description, which is invented by the naval architect. This translation gives meaning to numerical values attributed to input parameters and determines, for instance, whether a parameter varies hull length of a ship or the wingspan of an aircraft.

  The parametric model should allow input parameters to make large and concurrent changes to the entire geometric ship description of a service vessel (necessary to cover the many options outlined in Section 1.3). Unfortunately, the references in Section 2.3 illustrated that current parametric models can only make small changes to a single part of a geometric ship description (see Section 2.6).

Note that numerous parts of the parametric model do not undergo parameter-driven changes and, hence, remain static during the search process. The ship description is up to date once the parametric model has adjusted to new values of the input parameters. The resulting ship description is used as input for the second step: performance predictions.

- **Performance prediction tools**. The second step uses the resulting ship description to calculate its performances. This could concern, for example, the ship's weight and centre of gravity, its initial stability, resistance and powering characteristics or operability in waves. Performance prediction tools are human inventions, like the ship description they use as input. Furthermore, several performance prediction tools need to be combined in an integrated design model in order to identify conflicts between design requirements (see Sections 1.4 and 2.4). The performance predictions complement the ship description by assessing not only whether the relevant systems fit onboard, but also whether the attained performance levels enable their successful use at sea (see Section 1.2). Still, the output of performance prediction tools cannot be used directly, since search algorithms require concisely formulated ratings to guide the search process.
- **Calculate ratings**. The third step aggregates the performance levels into ratings, i.e., objectives and constraints, that the search algorithm uses to guide the search process. This is usually performed using some simple mathematical rules (always invented by humans). Section 2.4 highlighted some problems with formulating rules based on human opinions, such as those required for settling trade-offs between negotiable design requirements.

Together, the three steps ensure the translation of input parameters into ratings and thereby close the iterative loop shown in Fig. 3.2. In the process, a single ship description is generated, its performances predicted and its ratings calculated. As such, **performing one iteration generates one alternative** (which does not have to be feasible) of the many needed to cover the range of options the naval architect faces during the early stage design of service vessels.

The loop shown in Fig. 3.2 forms the basis of the search process, which is discussed below.

Running the search process

So far, the discussion of the loop developed in Fig. 3.1 and Fig. 3.2 considers a single iteration. This single iteration covers all steps needed to convert old input parameter values to new input parameter values. Obviously, performing a single iteration does not lead to design improvements, only to new parameter values.

Search algorithms therefore use multiple iterations to improve the ratings of a design ('to optimise', as it is commonly called), as shown in Fig. 3.3. Typically, a large number of iterations are used to profit from the speed increase offered by the available computational power combined with a lack of human interaction during the search process.

The three steps used in the search process are obvious: starting, running and stopping. Each is discussed below.

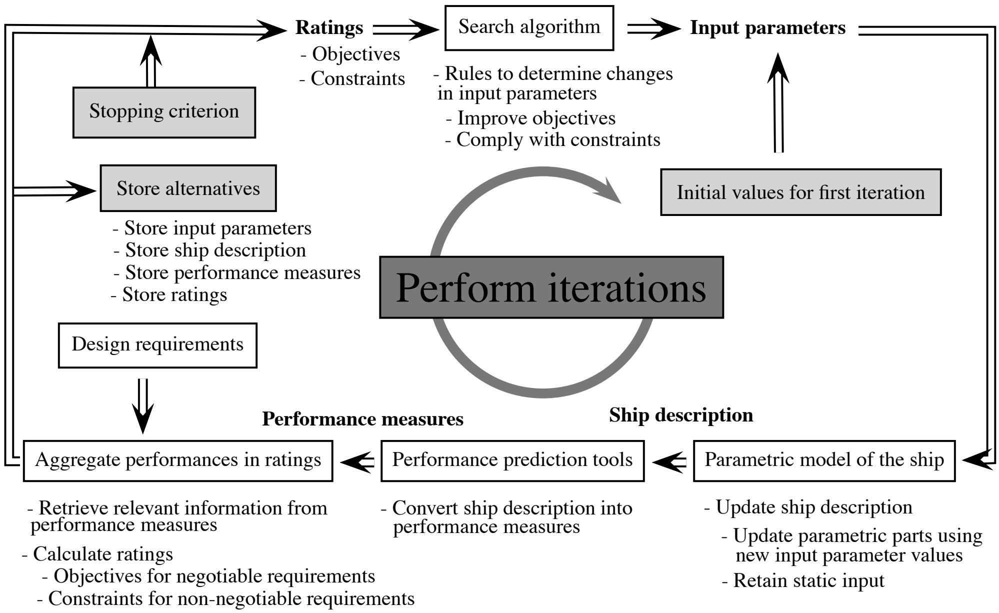


Fig. 3.3 – Fig. 3.2 updated with starting, running, stopping and storage

- **Starting**. The search process starts using a set of initial values for input parameters, e.g., created using a random number generator. This is necessary because no prior design (or designs) is available to calculate ratings. After calculating the ratings for these initial values, subsequent loops can be performed as discussed below.

- **Running**. Once started, the search process uses numerous iterations in order to search for designs with better ratings, using the steps shown in Fig. 3.3. The iterations may be performed sequentially on a single design, or concurrently on several designs, depending on the type of search algorithm being used. Note, both the number of iterations and number of designs considered during the search process are finite and defined by the naval architect.

- **Stopping**. At some point, the search process must stop to allow the naval architect to study the results. Various stopping criteria exist. For example, the process may stop because no further improvement of the ratings is possible, because a pre-defined number of iterations has been exceeded, or because a pre-defined time-span has passed. The latter is especially useful because it stops the search process the moment the results are required (this does not guarantee useful results, though).

Once stopped, the alternatives can be considered by the naval architect. To enable this, the alternatives need to be stored, which is discussed below.

Storing results

The search process uses a large number of iterations that adjust input parameter values to improve ratings. In the process, the search effort generates alternatives and predicts their performances. If these alternatives are not stored for later consideration, the search effort is meaningless.

For now, all information of all feasible alternatives found during the search process will be stored, i.e., the input parameters, the resulting description of the ship, its performances and the ratings. Section 3.5 discusses how its information is going to be used.

3.4 Issues encountered during use of search algorithms

Section 3.3 discussed how search algorithms might search for 'promising' alternatives during early stage ship design.

Unfortunately, one encounters several issues while trying to do so in practice. These issues are categorised according to the phase one encounters them in during the application of a search algorithm. The three phases are: preparing for running the search process, during the search process, and after the search process. The issues one encounters in each of the three phases are discussed in Sections 3.4.1 to 3.4.3.

3.4.1 Issues encountered when preparing for the search process

Fig. 3.4 shows that performing the search process requires a considerable number of human inventions. Of these, the parametric model of the ship and the performance prediction tools form the most important and time-consuming ones (ratings are invented as well, but issues related to their invention are discussed in Section 3.4.2). This role of human invention has three major consequences.

1. Both the parametric model and the performance prediction tools are invented by humans. They therefore constitute a considerable investment of human effort (Section 2.4) and to be worthwhile, they must be reusable to reduce human effort during several design projects. This means both the parametric model and the performance prediction tools should be able to cope with a wide range of service ship types, e.g., both dredgers and amphibious assault ships, defined at a suitable level of detail (see *Van der Nat [120]* and *Andrews [5]* for a discussion on the appropriate level of detail). Among other things, this requires the parametric model to handle different systems, i.e., support customisation of the set of systems aboard a service vessel. The same thing holds for performance prediction tools. It should be possible to add or remove prediction tools, if and when required, to make sure the relevant performances are predicted for the type of service vessel under design.

2. The parametric model must rely solely on changes in input parameter values to change the geometrical ship description. Any dependence on human interaction during the design iterations must be avoided to enable the application of search algorithms. Most importantly, the changes in input parameter values must lead to large and concurrent

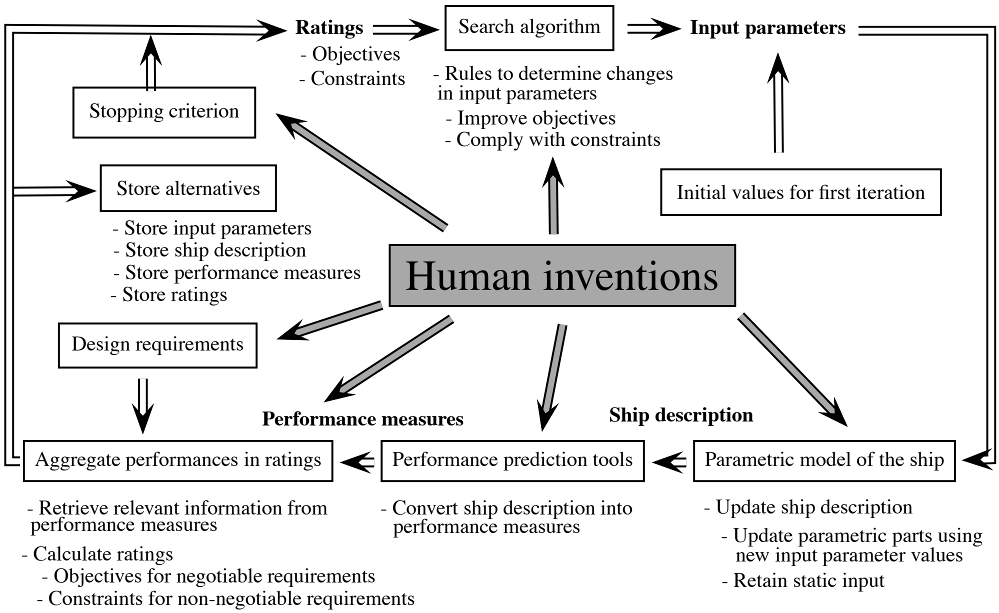


Fig. 3.4 – Role of human invention

changes to the entire ship description, so that the description can cover all options of interest (discussed in Section 1.3). Unfortunately, Section 2.6 identified that the references in Section 2.3 are unable to handle such large and concurrent changes to geometric descriptions.

3. The ability of input parameters to change the ship description is finite. The reason is that any independent change to the ship description -regardless of its magnitude- requires a separate input parameter, and the number of input parameters a search algorithm can use is finite. Only human invention can alter the number of input parameters used by a parametric ship description, as well as the bounds on the values these parameters may take (as discussed in Section 2.4.2).

These three issues have three consequences.

1. Not resolving the first issue means the effort reduction resulting from the application of search algorithms (proposed in Section 3.1) cannot be achieved. As a result, human effort will shift towards the development and alteration of the parametric model and the performance prediction tools in-between design projects.

2. Not resolving the second issue will see the generation of alternatives that only differ little and prevent the parametric model and search algorithm from generating the required large number of different alternatives required to cover the options outlined in Section 1.3. Again, human effort will shift towards the cumbersome adjustment of the parametric model and the performance prediction tools.

3. The third issue has the most important consequence: search algorithms cannot search outside the boundaries imposed by the parametric ship description (and the human inventions on which it is based). This does not mean that the parametric model cannot cover a wide range of alternatives. It does mean, however, that the **alternatives** will be 'unexpected' -at best- but **can never be 'novel'**, i.e., innovation remains a human premise.

3.4.2 Issues encountered when running the search process

One also encounters several issues after completing the preparation, when one actually performs a search with the search algorithm. These issues are discussed below. They are discussed in two groups: one related to yield and the other related to the 'lack of completeness' of ratings.

Yield

The search algorithm must find a sufficiently large set of relevant alternatives while considering a finite number of designs during the search process (see Section 3.3). This means the yield of the search process has to be sufficient. Unfortunately, one encounters three issues related to yield.

1. Search algorithms instigate changes by altering the value of input parameters. If one wants to change all parts of a ship description independently, one must use a large number of input parameters. Unfortunately, this simultaneously increases the number of combinations of input parameter values and thus the complexity of the search problem, which may reduce the yield.

2. Search algorithms require predefined rules to generate changes in input parameters (Section 3.3). These rules are invented prior to the search process and, crucially, do not change during the search process. Humans, in contrast, alter the design changes based on their experience and information gained from earlier design iterations, i.e., they can alter the rules with which to generate design changes during the search process (see *Simpson et al. [109]* for a brief discussion). This means that a search algorithm will be less efficient per design iteration compared to a naval architect, which can reduce the number of 'promising' alternatives it may find for a given number of designs considered.

3. Search algorithms perform a potentially large, but still finite number of design iterations. This means that they can only investigate a finite number of changes in parameter values, while the large number of input parameters means a vast number of combinations of input parameter values is available (typically far larger than the number of design iterations used during the search process).

The issues related to yield have three consequences.

1. Finding an insufficient number of alternatives means the naval architect does not have a sufficient number of alternatives to cover the large number of options under consideration during the early stage design of service vessels (Section 1.3). This may cause naval

architects to resort to rules of thumbs and assumptions, the very problem the design approach developed in this Chapter intends to address.

2. Focussing of the parametric model might be necessary to provide sufficient 'yield'. Such focussing steers the search process towards 'promising' alternatives, but does require human inventions, which, if not applied properly, can lead to a considerable increase in human effort. Moreover, excessive focussing will result in alternatives that look alike, i.e., will reduce diversity in the set of alternatives.

3. Given that the search algorithm performs a finite number of design iterations, it will be unlikely that any of the alternatives found will be considered 'perfect' by the naval architect. The reason is that, statistically, too few alternatives will be considered to have a reasonable chance of finding such a design (note, this assumes it is possible to guide the search effort towards such 'perfect' designs in the first place, which is discussed below).

   A simple example illustrates this. If one uses ten parameters with integer values between 1 and 10 to alter a parametric ship description, the search algorithm has 10^{10} combinations to consider. This number is far higher than the number of alternatives that can practically be considered during the early stage design of service vessels. This means that, from a probability point of view, there will be a large chance the design deemed 'perfect' by the naval architect will not be found during the search process.

Lack of 'completeness' of ratings

Section 3.3 explained how search algorithms use pre-defined ratings to guide the search process. There are two issues affecting the formulation of these ratings, and the way the search algorithm uses them.

1. The ratings the search algorithm uses are invented prior to the search process (Section 2.4 and 3.3). This takes effort and one must consider if it is worthwhile to develop such ratings. Whether it is, depends on the purpose of these ratings, i.e., whether they are used to pursue non-negotiable requirements or negotiable requirements (which were discussed in Section 1.4).

   - Formulating ratings for non-negotiable requirements is useful because there is general agreement on their non-negotiable nature. As such, they do not require alteration after their formulation or selection (i.e., they are not part of the 'wicked problem' discussed in Section 1.5). Most importantly, formulating them helps relieve the naval architect from the considerable effort needed to make the designs comply with these requirements.
   - Formulating ratings for individual negotiable requirements is also possible, as there is general agreement on how to predict and rate such individual requirements (speed is measured in knots, for instance).
   - Formulating ratings for trade-off decisions that settle conflicts between negotiable requirements is far more difficult. Section 2.4 discussed the implicit nature of such decisions, which are based on human opinions. These opinions will differ between different people involved in the design process and, most importantly,

these opinions are subject to revision depending on the impact the decisions have on the overall design of the vessel.

As a result, no firm agreement exists -at the start of the design process- on what constitutes the 'best' compromise and hence, any rule-based ratings formalising such compromises (for instance, by attributing numerical weight factors that indicate the relative importance attributed to different aspects of the design) may require frequent and extensive changes. These changes rely on human interaction (i.e., a dialogue between naval architect and customer to determine adjustments to the priorities) and could therefore reduce the speed with which design iterations can be performed.

Moreover, rule-based ratings formalising compromises will have little to no reusability, as different ships result from different compromises, and therefore require different ratings to define these compromises.

The implicit and thus provisional nature of trade-off decisions at the start of the design process, and their lack of reusability, caused the author to consider it not worthwhile to develop ratings that decide upon compromises between conflicting negotiable requirements a-priori. This differs from several of the approaches from Chapter 2, e.g., *Lee et al. [72]* and *Nick [88]*. Hence, an alternative approach is required both to identify conflicting design requirements, and to settle these conflicts by taking trade-off decisions.

Note, one may not have sufficient time available to develop all ratings one wishes to use within the scope of a design project, even when it is worthwhile to develop them.

2. The search algorithm can only consider a finite number of ratings (objectives and constraints) during the search process. This limitation has two related consequences.

    - First, the search algorithm will have a finite ability to distinguish between designs. For example, if two ratings are used: length and enclosed volume, the search algorithm cannot distinguish between designs with the same length and enclosed volume but different superstructure shapes. Humans on the other hand can identify such differences with ease (provided one does not limit the presentation of the ship designs to only length and enclosed volume but, e.g., also uses visualisation).
    - Second, the inability to distinguish between designs has a very important follow-on consequence. The inability means the search algorithm can only guide the search process towards designs with desirable characteristics (and away from erroneous designs) when ratings have been formulated to identify such designs. Crucially, any characteristic without a rating cannot be used by the search algorithm during the search process.

      A recent application by *Grigoropoulos et al. [55]* serves as an example. *Grigoropoulos et al. [55]* used a parametric hull form and a search algorithm to search for hull forms with reduced wave resistance and improved sea-keeping. As such, the search algorithm only used ratings for resistance and sea-keeping, but could not, and therefore did not, consider other performances relevant for hull form design, e.g., quality of the propeller inflow, during the search process.

      One could argue that this finite ability to identify promising and erroneous designs can be addressed by adding more prediction tools and ratings. Still, adding a single prediction tool, e.g., to predict propeller inflow in the example by *Grigoropoulos* *et al. [55]*, is not enough, as other performance predictions, e.g., manoeuvrability or slamming, will still be missing.

Hence, adding prediction tools and ratings will not prevent the search algorithm from generating designs with unsuitable performances, unless all relevant performances are predicted and included as ratings. Adding fewer prediction tools and ratings will merely change the performance that is unsuitable, e.g., from designs with unsuitable propeller inflow, to designs with bad manoeuvring characteristics.

Unfortunately, including all relevant prediction tools and ratings up to the point that they are as comprehensive (i.e., 'complete') as the judgement expressed by an experienced naval architect is very difficult. The reason is that the development and integration take so much human effort and time that it will be neither possible nor cost-effective to do so during early stage ship design.

This means that -in practice- the inability to distinguish between designs and its consequences can be reduced by increasing the number of ratings, but cannot be fully resolved.

The result is that -in practice- search algorithms cannot consider all aspects deemed relevant by a naval architect. As such, search algorithms are deemed to suffer from a 'lack of completeness' due to the incomplete set of ratings they must use to guide the search process.

The inability to distinguish between designs and the resulting 'lack of completeness' of ratings has two consequences.

1. The lack of completeness of the ratings guiding the search process means a search algorithm can very well focus on designs that naval architects would deem unsuitable, simply because the ratings the search algorithm needs to identify such designs as unsuitable are not formulated and, hence, lacking.

   The recent application by *Grigoropoulos et al. [55]* illustrates that the generation of such unsuitable designs does occur in practice. As discussed, *Grigoropoulos et al. [55]* only used ratings for resistance and sea-keeping, but did not consider other performances relevant for hull form design, e.g., quality of the propeller inflow, during the search process.

   The consequences of this lack of completeness (resistance and sea-keeping, but not propeller inflow) are shown in Fig. 3.5. The left image shows the hull form used as a

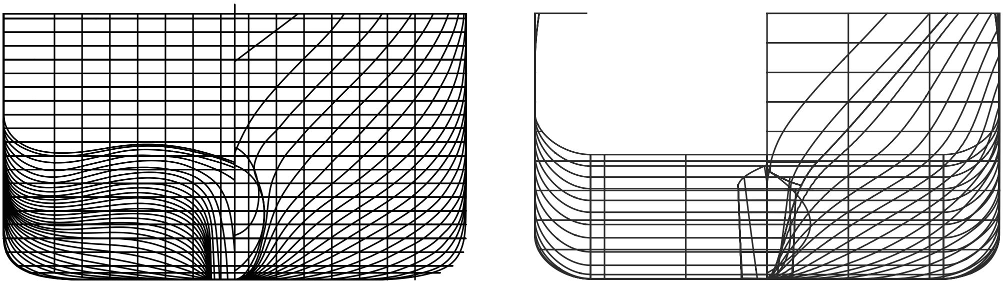


Fig. 3.5 – Original hull form (left) and hull form optimised for wave resistance and sea-keeping (right), taken from *Grigoropoulos et al. [55]*

starting point, the right image shows the 'best' hull form found after completing the search process. Comparison reveals that the original hull (on the left) has 'tunnels' in the aft ship, most probably to improve propeller inflow. After the search process, these 'tunnels' have disappeared (the right image in Fig. 3.5).

Though difficult to prove from the paper by *Grigoropoulos et al. [55]* alone, their disappearance most likely results from the fact that the search algorithm did not consider propeller inflow as one of the ratings, and exploited this unawareness to improve seakeeping and reduce resistance at the cost of unintentionally worsening propeller inflow.

This unintended behaviour of search algorithms causes them to generate designs considered 'too optimal' by *Buonanno and Mavris [22]*, i.e., designs that upon completion of the search process are 'optimised' only with respect to the ratings considered, and that simultaneously retain several basic flaws apparent to any engineer. These flaws affect characteristics not included in the ratings used by the search algorithm (e.g., as in the example in Fig. 3.5).

2. Section 2.4.3 discussed the implicit nature of trade-off decisions that settle conflicts between negotiable requirements. This, among other things, causes these decisions to be provisional until all relevant consequences they have on the overall ship design have become known.

   This implicit nature caused the author to argue that formulating ratings for settling compromises is very difficult, and not considered to be worthwhile. Their tendency to change once more insight is gained means the naval architect must adjust them repetitively, which does not contribute to reducing the human effort needed to perform design iterations.

   Not formulating ratings to settle trade-offs means exacerbating the lack of completeness of the ratings used to guide the search process. As a result, the search process will not be able to find a design that is considered by the naval architect to be the 'perfect' compromise between negotiable requirements, simply because ratings to identify such a design will not be formulated.

3.4.3 Issues encountered after finishing the search process

The search process generates a large set of alternatives (see Section 3.3). From this set, the naval architect has to identify and select a small number of promising alternatives for a more detailed investigation. These promising alternatives form a suitable compromise between the negotiable requirements and adhere to all non-negotiable requirements. One encounters several issues when trying to select such alternatives.

1. **Transparency**. Choosing a suitable alternative from the large set requires transparency to gain the insight as to why a particular alternative warrants further consideration.

2. **Information overload**. The large set of alternatives (with different system choices, configuration and performances) contains all information the naval architect needs to identify and select promising alternatives. Simultaneously, however, one will suffer from information overload, caused by the overwhelming amount of information to consider.

3. **Implicitness**. Selecting a small set of promising alternatives means taking trade-off decisions that settle conflicts between negotiable design requirements (Section 2.4). Their implicit nature ensures that these decisions remain provisional, i.e., subject to revision, until all relevant consequences have become known.

4. **Lack of completeness of ratings**. The lack of completeness of rule-based ratings (see Section 3.4.2), may cause ‘too optimal’ designs that suffer from basic flaws (see the example shown in Fig. 3.5). Moreover, it also limits the ability of the naval architect to distinguish between designs during the selection process.

5. **Acceptance**. The use of rule-based ratings led others (e.g., *Lee et al. [72]* and *Nick [88]*) to employ a **’best rating’ equals ’best design’** approach to help the human select a few promising alternatives. Unfortunately, this approach reduces the acceptance for the selected designs (Section 2.4), i.e., the willingness of humans to use the resulting design in the remainder of the design process.

   Two reasons are causing the lack of acceptance. First, using pre-defined ratings removes direct human involvement in the decision-making process and replaces it by the automated application of pre-defined decision rules. This confronts the naval architect with the ‘best’ alternative with the ‘best rating’, without giving him or her the option to validate decisions made by the search algorithm during the search process. Second, the lack of acceptance is increased by a lack of believability (*Andrews [4]*) caused by ‘too optimal’ designs containing basic flaws (caused by the lack of completeness of the ratings used to guide the search process, as Section 3.4.2 explained).

These issues have several consequences for the selection of a few promising designs from the large set of alternatives available after completing the search process. They are discussed below.

1. Transparency is essential for the selection of suitable designs. Without it, the naval architect is unable to make proper use of the large set of alternatives, e.g., to identify the necessary trade-offs between negotiable requirements or to establish how non-negotiable design requirements influence the choice of systems, configuration and performance levels.

2. Information overload will prevent the naval architect to make use of the information enclosed in the set of alternatives, simply because of a lack of insight. As such, preventing information overload forms a crucial element of providing transparency.

3. The provisional characteristics of trade-off decisions mean these decisions may need to be reversed or adjusted (hence, they cannot be taken a-priori, Section 2.4.3). Failure to provide the ability to revise them means one may take trade-off decisions and gain insight in their adverse consequences without having the ability to use this insight to address the adverse consequences of the decision by altering both the decision and the resulting design.

4. The erroneous designs caused by the ‘lack of completeness’ may cause the naval architect to draw erroneous conclusions from the set of alternatives. Moreover, one risks selecting such an unsuitable design for further use later on in the design process.

5. The ‘lack of completeness’ of ratings also makes it difficult to choose between different designs with similar ratings and, as a result, makes it more difficult to reduce the large set of designs to a few promising alternatives.

6. The lack of acceptance means naval architects are unwilling to accept the alternatives a search algorithm generates and, hence, are unwilling to use them in the design process. In that case, the application of search algorithms to enable the investigation of numerous alternatives is pointless, as none of the many alternatives found will actually be used.

3.5 The way forward

One might be tempted to think the issues identified in Section 3.4 are, in combination, sufficient reason to stop pursuing the application of search algorithms to improve the design process of service vessels.

The author thinks otherwise, and this Section proposes remedies for the issues raised in Section 3.4. The approach to remedy these issues builds upon the literature review in Chapter 2 and the subsequent discussions in Sections 3.1 to 3.4. The discussion is divided into three parts: alterations that improve the issues encountered prior to, during and after the search process. Each is discussed below.

3.5.1 Resolving issues encountered when preparing for the search process

Section 3.4.1 discussed the issues one encounters when preparing for the search process: the role of human invention and the resulting need for reusability, as well as the need to achieve large changes to the entire ship description by changing the values of input parameters.

Invention and reusability

Both the parametric model of the ship description and the performance prediction tools need to be invented (i.e., designed, following Section 1.6) by humans prior to the search process. Most importantly, these inventions need to be reusable in several design projects to recoup the human effort invested in them.

Fortunately, close consideration of the systems service vessels use indicates reusability should be achievable for the parametric model. For one, only a finite number of systems exists that can help the ship to fulfil her mission. Moreover, this finite availability is limited further by market availability and economic considerations (*O’Brien [92]* and *Greig et al. [52]*). A new warship, for example, will have a finite choice of weapons and sensors due to the fixed budget allocated in the defence procurement plan. For civilian vessels, some systems can be ruled out due to the negative impact they have on the profitability of the ship. More importantly, only a few systems will be truly novel while most other systems will differ only little between different types of service ships (a bow thruster in an offshore support vessel may be the same as one in an amphibious assault ship). Lastly, the development of new systems takes place prior to the early stage design of service vessels, due to the need for preparatory research and development (e.g., as in *Eefsen et al. [43]*).

As a result, the author considers there to be ample scope for **building a database of systems** that can be used in several design projects. In this database, the naval architect can store the spatial, weight and other properties of systems (see *Andrews and Dicks [7]* for a discussion of such attributes). When assembling a parametric model, one can thus select those systems relevant for the particular service vessel one is designing (these systems obviously need to be invented first). This approach also supports customisation, i.e., one can choose different systems based on their suitability, which Section 3.4.1 argued to be essential to achieve reusability. Earlier approaches that used such a collection of systems before assembling them in a bespoke ship design are found in (*Leopold and Reuter [73]*, *Andrews and Dicks [7]*, *Van der Nat [120]* and *McDonald [79]*).

A similar approach can be taken for performance prediction tools. One selects only those prediction tools relevant for the ship type at hand. For example, stability, resistance, and propulsion of mono-hulls will be relevant most of the time, whereas station keeping and vulnerability assessment to predict the impact of underwater explosions on a ship's hull will be most relevant for offshore vessels and for warships, respectively.

Still, some remarks regarding the possibilities of such a database are in order. Obviously, these systems stored in it must first be designed and, if necessary, be made adjustable for particular configurations (as discussed in Section 1.6). Therefore, the database should be able to support both the storage of simpler systems with fixed properties, as well of systems that rely on more elaborate sizing and positioning rules to describe their dependencies on the overall ship design, e.g., marine engineering systems (discussed in Section 1.6).

Note that the development of such sizing and positioning rules for systems is quite complex and received only moderate attention during the development of the novel design approach presented in this dissertation. The reason is that many of these rules are both ship type and system type dependent (witness the large effort required by *Van der Nat [120]* to develop accurate sizing and prediction models for the energy generation, storage and usage systems aboard diesel-electric submarines). A focus on developing ship specific system design rules would therefore dilute the effort to develop the novel whole-ship design approach. Such a shift in focus was considered undesirable and, hence, not pursued in the research presented here.

Still, Chapter 6 provides two examples of the ability to handle such more complex rules. First, it will present a rule-based application where a resistance and propulsion prediction is combined with a list of available diesel engines and gas turbines to enable a rough and initial sizing of the propulsion system to ensure the ship will meet the required speeds. Second, the same application uses a rule-set to position bulkheads while considering the presence of other systems and accounting for requirements prescribing the allowed degree of flooding when damaged (see the BSc. research project by *Van Diessen [122]* for details, which was performed under supervision of the author). Since these two examples are fairly limited, further effort to develop more rules to size and position complex, configuration-dependent systems is most certainly warranted.

Achieving large and concurrent changes using input parameters

The parametric model handles all design changes required to generate the large range of alternatives Section 1.3 argued to be essential for a proper investigation of promising combinations of system choices, configuration and performance levels. These design changes must be accomplished solely by value changes of input parameters, in order to benefit from the use of search algorithms (Section 3.2). Section 2.6 highlighted the problems the references discussed in Chapter 2 had in achieving this, and also observed that -in one reference: *Valdenazzi et al. [117]*- the limited changes were deliberately chosen to keep the problem manageable.

Two things must to be developed to enable large changes to the entire ship description using input parameters:

First, instigating large changes to the entire ship requires extensive changes to individual parts of the ship. These changes concern, e.g., shape and size of hull and superstructure, position and shape of decks and bulkheads and, finally, the number and type of systems as well as their shape, position and connectivity. Fortunately, the references in Chapter 2 illustrated how input parameters can change each part of a geometric ship description individually (often though with only minor changes).

Five examples of how input parameters can instigate relevant design changes are discussed below (based on Chapter 2). All types of input parameters: discrete, continuous and combinatorial discussed in Section 3.3 are used.

- Selection of ship systems. Search algorithms can use integer parameters to select systems from the database. For example, a value of one would result in using a medium speed diesel as a prime mover, whereas a value of two would result in a gas turbine. Discrete parameters can also remove systems from the parametric model.

- Number of systems. Search algorithms can use integer parameters to determine the number of systems onboard. For example, an input parameter can vary the number of cranes on a deep water drilling vessel between two and four.

- Position changes. The search algorithm can use input parameters that describe and alter the position (x, y and z-coordinates, for example), as well as the attitude of a system (e.g., a forward or transverse facing gun) in the ship's configuration. Several references from Chapter 2 used this ability to search for improved designs, e.g., *Smith et al. [110]* and *Lee et al. [72]*. Most importantly, all positions a system may take in the ship can be covered using only three, reusable coordinates per system (e.g., by using the ubiquitous x, y and z-coordinates).

- Shape changes. Continuous input parameters can alter the shape of systems, e.g., the hull or superstructure. Shape changes may be relatively large (*Peri and Dattola [98]*) or small (*Valdenazzi et al. [117]*). Moreover, when continuous changes prove insufficient, one can combine several parametric shape descriptions with a discrete parameter that determines which parametric description is used. For example, a value of one means using a parametric hull with a bulbous bow, while a value of two will see a parametric hull without a bulbous bow being used.

- Connectivity. Both discrete and combinatorial input parameters can alter the connectivity of systems (the latter by controlling the sequence in which connections are made to a larger network of connections). See *Asmara and Nienhuis [12]* for an example from the field of pipe routing.

These five types of parameter-driven changes enable input parameters to make large changes to all relevant parts of the ship description during the search process without requiring human interaction. It should therefore help to increase the number of alternatives one can consider during the early stage design of service vessels (as discussed in Section 3.3).

The 'unexpected, but never novel' issue from Section 3.4.1 remains, however. Fortunately, the ability to handle large and concurrent changes to the parametric ship description is expected to help expand the range of designs that can be covered by the description so that it should be able to encompass all alternatives of interest, at which point 'never novel' should no longer pose a problem.

Note, the use of input parameters proposed above is not considered definitive. Different use of parameters is entirely possible. In this dissertation, input parameters will be used to achieve large design changes that are often simple in nature (e.g., a change in hull dimensions by scaling a parent hull form), rather than investigating the 'best' parameterisation for each part of the ship description. The reason to do so is that it enables a focus on how to deal with interactions between multiple parameter-driven changes (which result from the integrated nature of ship design discussed in Section 1.4). The approach to deal with interactions between design changes is discussed below.

Second, a very important subsequent step is required. The **individual changes must be combined** in a single, coherent approach in order to achieve both **large and concurrent changes to the entire ship description**. For this, one must consider how the changes listed above interact with each other during the generation of alternatives. The integration of individual changes into a single approach is necessitated by the considerable interactions between system choice, configuration and performances discussed in Chapter 1. For example, if an input parameter alters the ship's beam, it may lead to unwanted changes in the position of systems onboard, as well as to changes in shape and size of the propulsion system caused by an increase in required propulsion power.

Dealing with such interactions between parameter-driven changes is proposed to take three steps: identification, resolution and prioritisation.

- **Identification**. One must identify whether and how one change affects or conflicts with other changes. For example, combining a hull shape change and a position change of the flight deck may result in a conflict between the two changes: 'insufficient width at the position of the flight deck'.
- **Resolution**. The conflict: 'insufficient width at the position of the flight deck', between hull shape change and flight deck position change can be addressed by altering either hull shape or flight deck position, or both. One has at least two options when considering such conflicts, as two or more parts of the ship are involved.
- **Prioritisation**. One must decide which of the changes of the resolution step to use. For the example, one must decide whether a hull shape change is more acceptable than a position change of the flight deck to address this particular conflict.

The identification, resolution and prioritisation steps are applied to all parts of the ship configuration to identify and remedy conflicts between design changes induced by input parameters.

Without it, the design changes generated by the search algorithm will not result in a coherent ship description, which makes the alternatives it generates useless.

In summary, it should be possible to make large changes to all parts of the ship description individually using only changes in input parameter values. The parametric model of the ship then accommodates these changes, such that it recognises conflicts between individual design changes, addresses them, and maintains a coherent ship description.

This ability to cope with such large and concurrent changes without human interaction is needed to enable the use of search algorithms to generate a large set of alternative service vessel designs (Sections 2.6 and 3.4.1). It will become one of the distinguishing features of the approach presented in this dissertation. It is developed in full in Chapter 4.

3.5.2 Resolving issues encountered when running the search process

Section 3.4.2 discussed the two issues one encounters during the search process: achieving sufficient yield and coping with a lack of completeness of the ratings used to direct the search process. Both are discussed below.

Achieving sufficient yield

The issues related to achieving sufficient yield (listed in Section 3.4.2) are resolved below. Yield is considered to be the number of relevant alternatives found during the search process (it can be expressed, for example, as the percentage of designs that is feasible relative to the total number of designs considered during the search process).

- The complexity of the search process can be reduced by lowering the number of parameters. For example, a considerable number of parameters can be removed, if the human manually selects the systems to use for a particular vessel. Similarly, many service vessels have configurations with considerable symmetry, which removes the need for parameters that alter a system's transverse position. Fewer input parameters equal lower complexity which could result in a higher yield.

- The yield of the search process can be improved by focussing, e.g., by setting appropriate bounds of input parameters to restrain the position of particular systems to useful locations in the ship. Section 3.4.2 argued that focussing should be applied with care. Without careful application, focussing may need to change frequently during the design process, which requires human interaction and slows down the design process as a result.

  Focussing should, therefore, not be used to simplify the development of the parametric model in Chapter 4. Instead, focussing must be considered as an adjustable feature to be added after developing the generic parametric ship description. Note, focussing can reduce the diversity in the set of alternatives, and one must strike a balance between sufficient yield through focussing and sufficient diversity on the other hand, as Chapter 6 will illustrate.

- The pre-defined rules a search algorithm uses will be less efficient (i.e., yield fewer useful designs per given number of iterations) than the design changes a naval architect generates. It is considered relatively difficult to resolve this issue by making pre-defined rules more efficient. Instead, the inefficiency of a search algorithm can be reduced by using an increased number of design iterations.

- Section 3.4.2 discussed that not each and every possible combination of input parameter values will be considered during the search process. Therefore, chances are that the alternative one is interested in might have some unwanted features, or even may not be generated at all.

  As a result, one has to accept that even the most promising alternatives resulting from a search by a search algorithm can be improved further. Fortunately, the approaches discussed in Chapter 2 that rely on extensive human interaction, e.g., *Andrews and Pawling [8]* or *Van Oers et al. [131]*, can make the few, dedicated changes with ease. The alternatives considered for such refinement preferably have a bit of 'room (margin) for changes' such that these design changes do not lead to large changes to the overall ship configuration, i.e., the alternatives should form a robust starting point. Note, such changes take time and cannot be made to every alternative; one must therefore select the most promising alternatives that require the least amount of change.

Resolving with the 'lack of completeness'

Section 3.4.2 discussed the problems one encounters in the formulation of ratings that steer the search process. The problems ensure the search algorithm is unable to identify and consider all characteristics relevant to the naval architect, i.e., the search process is guided by an incomplete set of ratings. This 'lack of completeness' may lead to 'too optimal' designs, and results in the inability to identify, and hence find, a compromise between negotiable requirements a naval architect would deem 'perfect' (due to the implicit nature of compromise decisions).

Section 3.4.2 already discarded an increase in the number of ratings as a solution. Instead, the 'lack of completeness' of ratings will be resolved using three different steps.

1. If a lack of completeness leads to 'too optimal' designs (with basic flaws), one could consider designs that are 'optimal' in the normal sense of the word (without basic flaws) to be 'suboptimal' (see *van Oers et al. [130]* for some examples). This means that one can remedy the lack of completeness -to some extent- by considering all feasible alternatives found during the search process, both 'optimal' and 'suboptimal'.

   This remedy is possible as a lack of 'completeness' of ratings **does not prevent** the search algorithm from generating alternatives that are 'promising' with regard to characteristics not included in its ratings. It merely **does not recognise** them, making adjustment of the search process impossible (which will probably reduce yield). Most importantly, the ability to generate such designs depends solely on the parametric model and the input parameters it uses.

2. The lack of completeness means the naval architect has some freedom to choose which ratings are used by the search algorithm. Ratings for non-negotiable and negotiable requirements are discussed separately.

   - Section 3.4.2 argued that one should focus on formulating constraints that safeguard non-negotiable design requirements. The reason is straightforward, they do not change during the search process and using them as constraints enables the search algorithm to consider them without human interaction. This reduces the effort used to safeguard these requirements, and enables the naval architect to focus on settling trade-offs. Note, such constraints should be calculated with considerable accuracy, as the presence of errors in their calculation cannot be corrected without using human intervention.

Moreover, Section 3.4.2 argued the time and effort available to develop such ratings is limited and that one may not have all ratings one needs. Given the limited time available, the development of these ratings should therefore focus on formulating a set of constraints that cover the most important non-negotiable requirements, i.e., that have a large impact on the ship's configuration. For surface vessels, this can be the ability to float in upright stable condition, for instance. Focussing on the important non-negotiable requirements should ensure **the design is 'not wrong'**.

Ratings for any non-negotiable requirements left unformulated will be addressed by considering all feasible alternatives found during the search process, including those with some extra 'room / margin for error' (as mentioned in the discussion of ways to improve yield). This will help resolve the overall 'lack of completeness' issue after completing the search process (see Section 3.5.3).

- Section 3.4.2 explained one cannot search directly for 'perfect' compromises between negotiable requirements due to their implicit nature. It therefore is worthwhile to consider the consequences that different compromises between negotiable requirements have: different compromises will lead to different designs of the ship. Crucially, this observation can be reversed: if designs differ, they must represent different compromises.

  As such, the need to formulate ratings defining the 'best' compromise can be removed, provided one can ensure that the large set of alternatives is diverse enough (to be discussed in Chapter 5.4.1).

The proposed use of ratings enables the search algorithm to search for alternatives that meet a basic set of non-negotiable requirements. Simultaneously, it will ensure the alternatives differ and thus reflect different compromises.

Several benefits follow from this. For example, it reduces the number of ratings to be invented, for no ratings are used to search for the 'best' compromises between non-negotiable requirements. In addition, it facilitates reusability. The ratings for non-negotiable requirements can often be reused directly, e.g., many types of service ships share a need to float upright in stable condition (though the relevant stability criteria will depend on the particular type of service vessel).

Reusability of ratings for settling conflicts between negotiable requirements is achieved by removing the need to formulate ratings defining the 'best' compromise (which differ between service vessels) and by providing diversity to cover a far wider range of trade-offs, instead. Still, the disadvantage of using diversity is that the search algorithm's yield (the number of useful designs found) could drop considerably, while achieving sufficient yield is already an issue (as discussed above).

Note that this proposed use of diversity instead of ratings that settle conflicts between negotiable requirements was developed over the course of applying the design approach developed in this Chapter. The results of these test cases (each with different ratings) were reported in *Van Oers et al. [129]*, *Van Oers et al. [132]*, *Wagner [135]* and *Van Oers et al. [133]*, respectively.

3. The lack of formulation of some ratings does not mean the aspects they concern are unimportant. For example, one must still identify 'too-optimal' designs to prevent their selection after the search process. This means an alternative approach is necessary to complement rule-based ratings.

   This approach is based on close consideration of how search algorithms use ratings to alter the value of input parameters. The only thing considered in this process is their value itself, and not the way this value is calculated. This has let several authors, e.g., *Kim and Cho [67]* and *Buonanno and Mavris [22]*, to replace rule-based ratings by humans who attribute numerical values to ratings after a visual inspection of one or more designs. This removes the need to invent rule-based ratings prior to the search process and replaces them with ratings based on human engineering judgement.

   This approach does have drawbacks, however, as using human interaction during the search process forms a considerable bottleneck (Section 3.3). More specifically, studying alternatives and rating them takes time, time during which the search algorithm has to wait until the ratings have been decided upon. Moreover, the number of alternatives considered must be small in order to remain comprehensible to humans, which limits the number of alternatives the search algorithm can consider during the search process. Lastly, rating alternatives repetitively is a tedious job that gets boring rather quickly. Despite these drawbacks, the use of human engineering judgement does resolve the lack of completeness, but slows down the search process as a result.

   Fortunately, removing the drawbacks of interactively rating designs during the search process is straightforward: humans will rate designs using visualisation **after** completing the search process, which removes the above-mentioned bottlenecks. This is a simple way to remedy the lack of completeness. Note, the choice to use human engineering judgement after the search process entails that the set of alternatives is static and cannot be expanded with new designs that benefit from the judgement expressed.

3.5.3 Resolving issues encountered after finishing the search process

Section 3.4.3 discussed five issues one encounters when selecting promising alternatives. These issues are: transparency, information overload, implicitness, the lack of completeness (discussed in Section 3.4.2) and acceptance. Each of them is addressed below.

1. Transparency. Selecting promising configurations requires insight in why they should be considered promising. Promising, however, is a relative rating, which is measured against other designs in the set, including those considered 'unpromising'. Therefore, one can have transparency, i.e., learn why a particular configuration warrants further attention, only by investigating all relevant aspects of all feasible configurations found during the search process.

2. Information overload. Section 3.4.3 argued that the overwhelming amount of information in the large set of alternatives is useless without the means to use it properly. Information overload must and will be prevented using filtering, i.e., by considering only part of the wealth of information available at a time. Filtering must be adjustable to investigate different aspects of all feasible ships (e.g., view only designs with low resistance, versus designs with low cost). Moreover, one should be able to combine multiple filters,

e.g., show designs with both high speed and low cost, to identify trade-offs. Lastly, filtering can be applied to all aspects of all feasible designs, such that no information is left unconsidered unintentionally.

3. Implicitness. Section 2.4 and Section 3.4.2 discussed trade-offs between negotiable requirements and their implicit nature. This implicitness must be dealt with in order to support the naval architect when settling conflicting negotiable requirements. The implicit nature of a compromise decision ensures that it remains provisional until all its relevant consequences are known. This means that supporting the investigation of different compromises entails increasing the speed with which their consequences become known, to help establish whether there is a need for the trade-off decisions to be adjusted.

   Fortunately, the availability of a large and diverse set of alternatives enables just that. Normally, a compromise decision is taken prior to generating an alternative (a-priori, Section 2.4.3), i.e., it forms the basis for further work. Working out its consequences takes time and, therefore, delays feedback on the degree of adjustment the compromise decision requires to resolve its implicitness.

   Having a diverse set of alternatives with a full description of all systems, configuration and performances simplifies working out the consequences of a compromise decision considerably, by using an a-posteriori approach (Section 2.4.3). Implicitness is resolved in a rapid test-and-reconsider cycle. It starts by removing (i.e., filtering out) any alternatives that do not comply with the compromise decision and studying whether the characteristics of the remaining alternatives are acceptable, before reflecting upon the appropriateness of the decision. The large and diverse set of designs will ensure that a broad range of compromises can be considered.

   The finite number of alternatives the search algorithm evaluates means the resulting set of alternatives cannot contain an alternative for every different compromise between negotiable requirements. Still, one can adjust the parametric model of the ship and the ratings and then repeat the search process to fill in any such blank spots.

   This approach to work out the consequences of compromises between negotiable requirements facilitates the rapid investigation of different compromises and their consequences. As such, it offers an efficient way to remove the implicit nature of compromise decisions.

4. Lack of completeness of ratings. Addressing the lack of completeness of ratings was already touched upon in Section 3.5.2. It is fairly straightforward. The lack of completeness may lead to erroneous designs, with obvious flaws. Identifying these flaws and removing erroneous designs from the set of alternatives requires some form of visualisation familiar to a naval architect, i.e., a general arrangement plan. A familiar context also helps the naval architect to distinguish between designs based on engineering judgement, enabling aspects left unconsidered by the search algorithm to be included after the search process, thereby reducing the need to formulate a complete set of rule-based ratings.

   One must consider information overload when addressing the lack of completeness using visualisation, as reviewing a large number of general arrangement plans quickly becomes cumbersome. The filtering approach should therefore be extended to enable the filtered visualisation of multiple geometric ship descriptions concurrently, in addition to being able to filter numerical aspects of designs.

5. Acceptance. Section 3.4.3 explained how search algorithms fail to create acceptance, i.e., are unable to instil a sense of ownership for the alternatives that are created. Two of the causes for the lack of acceptance: the lack of completeness of ratings, as well as the implicit nature of compromise decisions were addressed already.

   The remaining issue is the 'best rating equals best design' approach used by many references from Chapter 2 to select promising designs. This approach has two problems related to acceptance: first, a lack of insight why the 'best rating' indeed is the 'best alternative' and, second, the removal of decision-making responsibility of the naval architect through the use of pre-defined rule-based ratings during the search process.

   Resolving the first problem is simple. Enable the naval architect to consider all feasible alternatives found during the search process on all relevant aspects (regardless of the values of their ratings) helps him / her to learn why some alternatives are to be preferred over others (i.e., the best way to see why designs are 'better' is by comparing them with 'worse' designs). This is supported by the same approach as used to address the lack of completeness of ratings.

   The second problem is resolved by returning part of the decision-making responsibility to the naval architect. The division of decision-making responsibility is based on the negotiability of ratings (discussed in Section 3.5.2).

   - The decision-making responsibility related to non-negotiable requirements resides predominantly with the search algorithm, as rule-based ratings can be formulated for many non-negotiable requirements (such as the ability to float). Such ratings can be applied directly to focus the search process towards alternatives that comply with these non-negotiable requirements because there is general agreement on their definition.

   - The decision-making responsibility related to negotiable requirements resides solely with the naval architect. The reason is straightforward. Section 3.4.2 argued it to be not worthwhile to formulate ratings that settle compromises, due to their implicit nature and their reliance on human opinions that vary considerably.

     Hence, the lack of such ratings makes it impossible for the search algorithm to take trade-off decisions during the search process and therefore, decision-making responsibility must be handed back to the naval architect.

     More importantly, this responsibility helps the naval architect to form well-founded opinions after thoroughly investigating different trade-offs. Given that the naval architect is responsible for both the investigation and decision-making means the acceptance is generated instantly, i.e., one is unlikely to object to one's own opinions and decisions. Generating a large and diverse set of alternatives facilitates this investigation, as the diversity offers a wide range of different trade-offs to consider (Section 3.5.2).

     Lastly, using human engineering judgement, triggered by visualising designs in a familiar context, helps one to rate and select designs based on characteristics not included in the pre-defined ratings that guide the search process. Expressing human judgement thus will resolve the lack of completeness and, as a result, increase the quality of designs by identifying and removing those designs retaining basic flaws. As such, human judgement helps increase believability of the results (see *Andrews [4]*), which in turn increases one's willingness to accept them.

Summarised, resolving the five issues will support the naval architect in the transparent investigation of the large and diverse set of alternatives. It helps prevent information overload, and will enable a rapid investigation of different trade-offs to deal with their implicit nature. Visualisation and human engineering judgement will address the lack of completeness and its consequences, while studying all feasible alternatives found during the search process provides transparency, as it illustrates why specific alternatives should be considered to be preferable.

Most importantly, returning responsibility for taking trade-off decisions to the naval architect helps generate acceptance, while formulating rule-based ratings enables the search algorithm to search for alternatives that comply with a basic set of non-negotiable requirements. The latter enables the naval architect to focus his efforts on investigating and settling trade-offs between negotiable requirements (for which no ratings are formulated), instead of bearing the burden of ensuring the ship will float in stable upright condition, for instance.

3.6 Outline of the approach and accompanying design process

Sections 3.1 to 3.5 discussed the potential ability of search algorithms to reduce the effort required to generate a large number of alternative ship designs. They also discussed a whole range of issues that must be resolved to realise these benefits in practice.

This Section combines the lessons from the previous Sections in one coherent design process. Fig. 3.6 shows the overall outline of the approach. It combines human interaction with the application of a search algorithm and a geometric ship description (able to undergo large parameter-driven changes to the entire geometrical description) to generate a large and diverse set of service vessel designs, from which the naval architect can choose the most appropriate design.

The approach consists of several interconnected design loops (it is a more elaborate version of Fig. 2.1). Furthermore, it is an expanded version of the generic process used by search algorithms (shown earlier in Figs. 3.1 to 3.4). The modifications follow from resolving the issues as discussed in Section 3.5, which was necessary to enable the practical application of search algorithms in early stage ship design.

Each of the steps in Fig. 3.6 is discussed below.

1. Preparation comes first. It concerns inventing and implementing the parts of the parametric ship description, the performance prediction tools, the ratings, and the search algorithm. They are subsequently stored in the database in preparation for the second step (Sections 3.4.1 and 3.5.1).

2. From the database, humans can assemble a bespoke parametric model using the systems relevant for the type of service vessel under consideration. In addition, one can link up a set of performance prediction tools and accompanying ratings and combine it with a search algorithm to develop an integrated design model that checks compliance with the relevant design requirements. The database approach ensures reusability of the parametric ship description, the performance prediction tools, ratings and search algorithm (Sections 3.4.1 and 3.5.1).

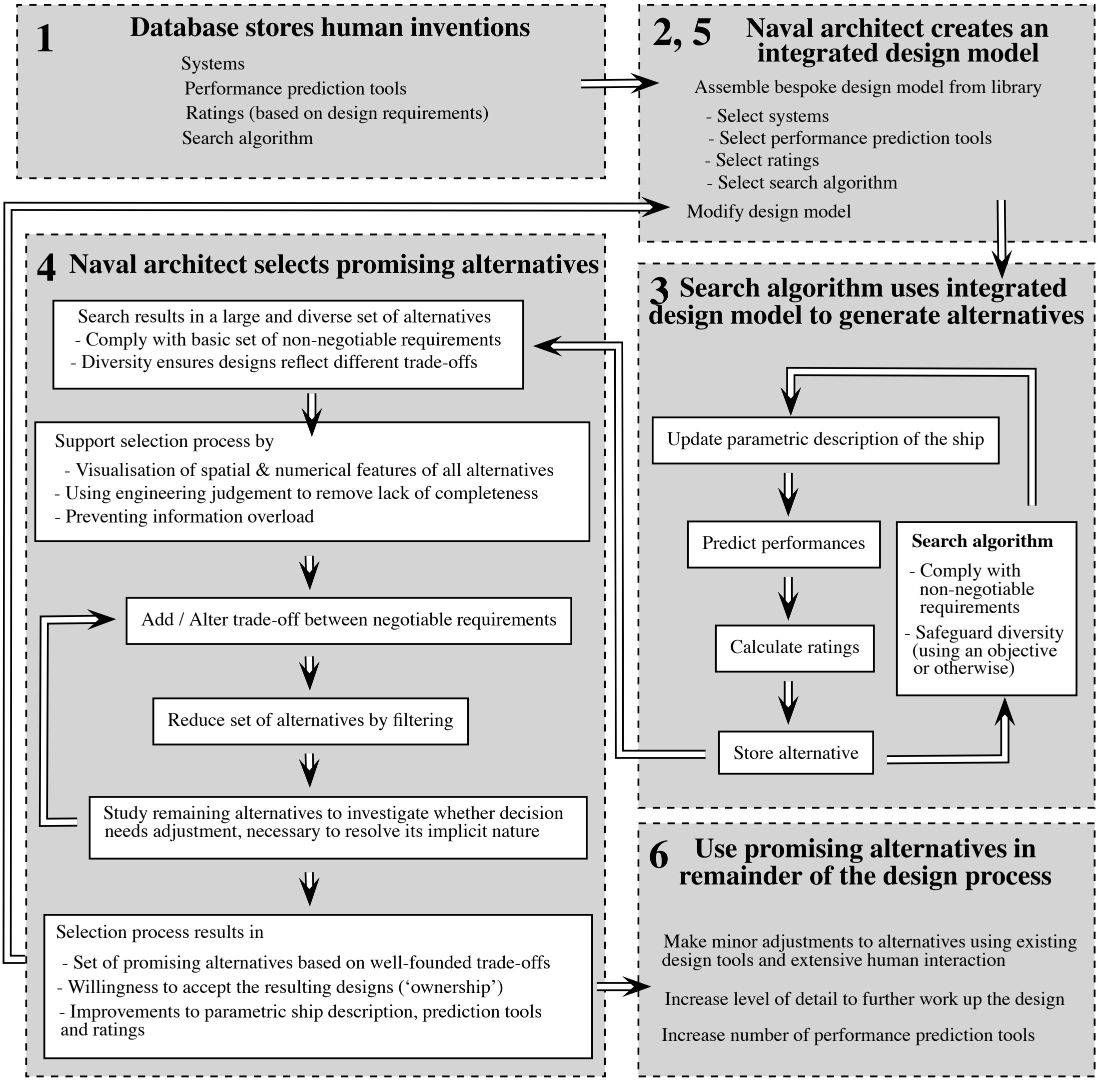


Fig. 3.6 – Outline of the approach for the early stage design of service vessels (numbers refer to the steps discussed in Section 3.6)

3. Once assembled, the search algorithm uses the parametric model, performance prediction tools and ratings to generate a large and diverse set of alternatives that comply with the non-negotiable design requirements (formulated as constraints). The ability to achieve large and concurrent changes to the entire ship description (Section 3.5.1) ensures the alternatives can cover the range of options one wants to consider (Section 1.3).

4. The naval architect investigates all feasible alternatives found in the search process in a transparent manner. This helps to gain insight in the need for trade-offs between negotiable design requirements and learn how requirements, system choices, configuration and performances interact (Section 1.5). Visualisation and filtering help to create this

insight, and help to trigger the expression of engineering judgement needed to distinguish between designs on aspects for which no ratings were formulated (Section 3.5.2 and 3.5.3). It also enables the identification of erroneous designs caused by a lack of completeness (Section 3.5.2).

Ultimately, the naval architect uses this insight to select a small set of promising alternatives that form a suitable compromise between negotiable requirements. The naval architect is responsible for selecting designs, which generates the crucial acceptance for the outcome of the selection process (Section 3.5.3).

5. The investigation and selection process may give rise to alterations to the parametric ship description, the performance prediction tools and the ratings used in the search process. For example, earlier studies may have revealed a mono-hull to be unsuitable. However, investigating a multi-hull design requires a change in stability and resistance prediction tools in order to make the proper predictions. Also, one may use different prediction tools for the same type of performance prediction, e.g., to increase accuracy or to study the impact on the overall ship design. Humans can make such alterations, if required, and study their impact by repeating steps 2 to 4.

6. The investigation and selection process should lead to a sufficiently interesting and acceptable design that will be considered in more detail (and refined if necessary). This can be achieved using existing design approaches discussed in Chapter 2, e.g., *Andrews and Pawling [8]* or *Van Oers et al. [131]*.

Together, the steps in Fig. 3.6 should exploit the ability of search algorithms to reduce the effort required to investigate numerous alternatives during the early stage design of service vessels. Section 3.7 concludes this Chapter by discussing the extent with which each step contributes to the reduction in effort.

3.7 Conclusions

One issue remains. After the elaborate discussion of their benefits and drawbacks presented in Sections 3.1 to 3.5, one might wonder whether the application of search algorithms as proposed in Section 3.2 will reduce the effort required to investigate a large number of alternative ship configurations during the early stage design of service vessels. Such a reduction of human effort is crucial for attaining the research objective outlined in Chapter 1.

The hope is that the complete approach shown in Fig. 3.6 will be able to reduce this effort, for reasons summarised below. The distribution of the effort reduction varies, however. Sometimes, effort reduction is possible within the scope of a single design project, while in other cases, it requires several design projects. In one case: the selection of designs, it is necessary to increase effort first to realise larger reductions later on in the process.

- The human effort required to invent parts of the ship's description cannot be reduced. It can be made worthwhile, however, by ensuring reusability over several design projects. Reusability is enhanced further by using a suitable breakdown that supports customisation of the parametric ship description for different types of service vessels.

- The application of a parametric ship description that is able to achieve large changes to the entire ship description by solely using input parameters also reduces human effort. It removes the need for the inventions normally required to handle these changes during the generation of alternatives.
- The decision to use search algorithms to search for alternatives that meet a basic set of non-negotiable requirements (formulated as constraints) relieves the naval architect from the burden of ensuring compliance with these requirements. It reduces human effort considerably as, in practice, one tends to focus first on meeting non-negotiable requirements, before considering negotiable ones.
- The choice to use search algorithms to search for different alternatives provides the naval architect with a range of designs that reflect different compromises. It thus reduces considerably the need to invent ratings to cover each and every possible compromise one might encounter.
- The availability of a set of alternatives with different system choices, configurations and performance levels that meet non-negotiable requirements helps the naval architect to get rapid feedback on different compromises in a transparent manner. Working out the consequences of the trade-off can be performed in a simple and rapid way by filtering the set of alternatives to gain insight in its suitability, before settling on the final decision. It thus reduces the effort required to address the implicit nature of compromise decisions.
- The use of visualisation and engineering judgement during the investigation of alternatives helps the naval architect to identify 'too optimal' designs, based on one's own judgement. It reduces human effort, as one does not have to formulate all ratings required to prevent such flaws. Moreover, visualisation allows one to distinguish between designs, further reducing the effort required to formulate rule-based ratings prior to the search process.
- Human responsibility for taking trade-off decisions helps generate acceptance for the results. As such, it does not reduce human effort directly; rather, it increases it. By doing so, however, it enables the practical acceptance of the alternatives generated by the search algorithms. On balance, this extra investment of human effort will be compensated by the speed with which the search algorithm can generate alternatives.

The reasons outlined above illustrate how -on balance- the approach proposed in this Chapter should reduce the effort required to investigate numerous alternatives during the early stage design of service vessels. This indicates that search algorithms can be used successfully to improve the speed of design iterations (as suggested in Section 3.2) even while accounting for the many issues that must be resolved before one can apply them in practical ship design efforts.

The resulting design process in Fig. 3.6 should be able to generate a large number of different ship designs (with different system choices, configuration and performance) without resorting to extensive human interaction during the search process, while ensuring the designs comply with non-negotiable design requirements. This large, diverse and feasible set of designs enables the naval architect to investigate different trade-offs between negotiable requirements. The resulting insight should help fuel the dialogue between owner and ship designer that is so important to address the 'wicked' interaction between the ship's design and the design requirements.

In summary, with the approach, the naval architect can thoroughly investigate the many options before identifying and selecting those combinations of systems, configurations and performance levels considered most promising. Most importantly, he or she can do so without investing a prohibitive amount of effort (for the reasons outlined above), which means the design approach presented in this Chapter should be able to address the problems outlined in Section 1.5.

What remains is an introduction of the remaining Chapters of this dissertation. Chapter 4 will develop the parametric ship description, while Chapter 5 discusses the search algorithm, prediction tools and ratings that will be used during the search process. Chapter 6 illustrates the possibilities of the approach developed in Chapters 3 to 5 by showing two applications: a deep water drilling vessel and a frigate. Chapter 7 outlines the development of the approach to support the naval architect during the investigation of, and selection from, the set of alternatives resulting from the applications in Chapter 6.

Chapter 4

Describing and altering ship configurations

4.1 Introduction

Chapter 3 outlined the novel approach intended to support the naval architect during the early stage design of service vessels. It uses a search algorithm combined with a parametric ship description to generate a large and diverse set of alternatives that comply with non-negotiable design requirements. This set helps the naval architect to thoroughly investigate the many options (Section 1.3), before selecting a suitable alternative using the insight gained from the investigation (Section 3.6).

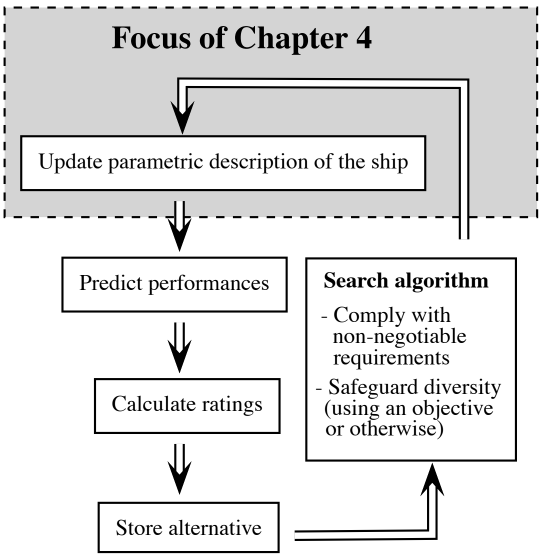


Fig. 4.1 – Focus of this Chapter

The parametric ship description lies at the core of the new approach (Fig. 4.1 shows the relevant part of Fig. 3.6). No alternatives can be generated without it. To enable the successful generation of a large and diverse set of alternatives, the parametric ship description must both address the issues Sections 2.6, 3.4 and 3.5 outlined, and integrate the beneficial trends discussed in Section 2.5.

More specifically, it must have the following characteristics.

- Describe the entire ship at a level of detail deemed suitable by the naval architect. The need to describe the entire ship is obvious, while Section 2.6 explained different ship design problems can require different levels of detail to address them (citing *Van der Nat [120]* and *Andrews [5]*), requiring the parametric ship description to handle a variable level of detail.

- Ensure reusability for various ship types. Section 2.4.2 argued inventions such as the parametric ship description form a considerable investment of human effort. To be worthwhile, the parametric model should be reusable and applicable for the design of a broad range of service vessels.

- Handle large changes to the entire ship definition introduced by changes in input parameter values. Section 2.6 illustrated most parameter-driven changes used by existing approaches result in fairly minor changes to the ship description. To be applicable for early stage ship design, the magnitude of these changes should increase.

- Apply these changes concurrently to the entire ship description, i.e., hull, superstructure, decks, bulkheads and systems. Parameter-driven design changes should not only be large, but should be also applied to the entire ship description concurrently. Unfortunately, Section 2.6 highlighted that the ability to make such parameter-driven changes is lacking in existing approaches.

- Support a variable degree of focusing by the naval architect. Section 3.5.2 argued the search process must result in a sufficient number of alternatives and may require focussing to do so. Still, the degree of focussing will depend, most likely, on the type of service vessel being designed and should therefore be adaptable.

The ship description will be developed from the ground up to attain these characteristics. Its development starts in Section 4.2 by outlining arguments why the early stage design of service vessels should be considered as an example from a broader class of mathematical problems called ‘packing problems’.

4.2 Packing as a model for ship configuration design

The successful development of the parametric ship description in this Chapter first and foremost requires an appropriate model which should capture correctly how naval architects create and alter the entire ship description in practice.

Two observations form the starting point for the model; they are listed below.

- First, the size of any ship is finite, meaning she can be enclosed in a box of finite dimensions (as in Fig. 4.2).
- Second, systems cannot occupy the same space at the same time, generally speaking. For example, a gun on a warship that overlaps with a helicopter deck will interfere with flight operations (as in Fig. 4.3).

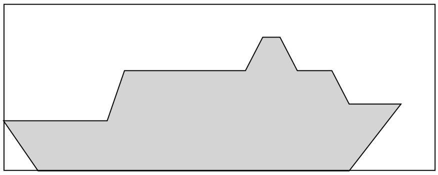


Fig. 4.2 – Ship enclosed by a box of finite size

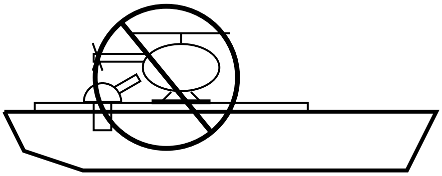


Fig. 4.3 – Unwanted overlap between a gun and a flight deck

Both observations identify key characteristics of a broad class of mathematical problems called **packing problems**. Packing problems come in many different forms (see *Dyckhoff [39]* for an overview) and the observations in Fig. 4.2 and 4.3 point towards a spatial or geometric packing problem.

Geometric packing problems deal with placing several geometric objects such that:

- **All objects overlap completely with a larger positioning space**, e.g., the ship enclosed by the box in Fig. 4.2
- **All objects prevent unwanted overlap among themselves**, e.g., avoid the situation shown in Fig. 4.3

Figs. 4.4 to 4.6 show a simple three-dimensional geometrical packing problem as an example from *Dyckhoff [39]*. A single large block (Fig. 4.4) is used as a positioning space in which several smaller blocks (Fig. 4.5) are packed without overlap among them (Fig. 4.6).

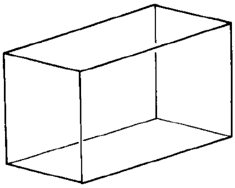


Fig. 4.4 – Positioning space

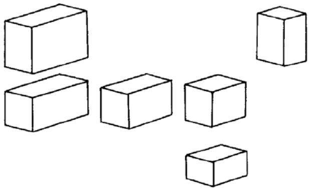


Fig. 4.5 – Objects

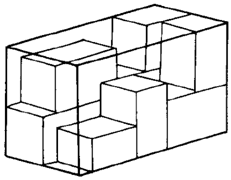


Fig. 4.6 – Resulting configuration

At first sight, the decision to use packing as a basis for the parametric ship description on the two very simple observations seems somewhat fanciful and some additional insight in the soundness of the decision is required. This is provided by comparing the simple packing problem shown in Figs. 4.4 to 4.6 with a more complex, ship-related packing problem.

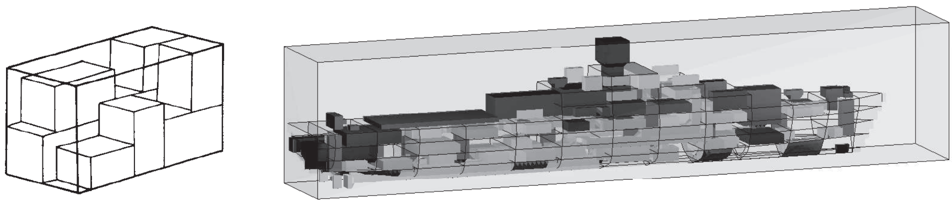


Fig. 4.7 – Comparison between a simple packing problem from *Dyckhoff [39]* (left) and a complex ship-related packing problem from Chapter 6 (right)

Fig. 4.7 provides the comparison and highlights the similarities: the objects may differ, overlap rules may differ, but the key problem remains: packing objects in a positioning space while complying with overlap rules. Moreover, the two observations made earlier are valid for any type of service vessel, regardless of size and systems and thus support reusability (one of the requirements outlined in Section 4.1).

Further development is necessary to ensure a packing-based description meets the requirements laid out in Section 4.1. Section 4.3 starts the development by discussing four recent packing applications considered during the development of the packing approach presented in this Chapter. Section 4.4 subsequently introduces the requisites of spatial packing problems before discussing their integration in a coherent packing approach in Section 4.7 and subsequent Sections.

4.3 Four relevant packing references

This dissertation does not provide a full literature review of existing packing approaches. Instead, four recent and relevant packing approaches from Section 2.3 are discussed in more detail. They serve either as inspiration, or as examples to be improved upon. Furthermore, additional references are used throughout this Chapter at appropriate steps in the development of the ship description.

4.3.1 University College London (UCL)

UCL researched the early stage configuration design of ships since the eighties (*Andrews [2]* and *Andrews [3]*). This resulted in the 'Functional Building Block' approach (*Andrews and Dicks [7]*) and its implementation in *Paramarine / Surfcon* (*Andrews and Pawling [8]*, *Pawling [95]*). The approach was applied to a range of ship types, e.g., frigates (*Andrews et al. [11]*), aircraft carriers (*Andrews [6]*), submarines (*Andrews et al. [10]*) as well as civilian vessels.

UCL's approach has five important features relevant for the packing approach developed in this Chapter. First, the packing approach may use 'Functional Building Blocks' to represent the ship's systems; a topic Section 4.4.1 elaborates upon. Second, the approach handles a wide range of ship types, offering reusability (Section 3.5.1). Third, UCL combines numerical

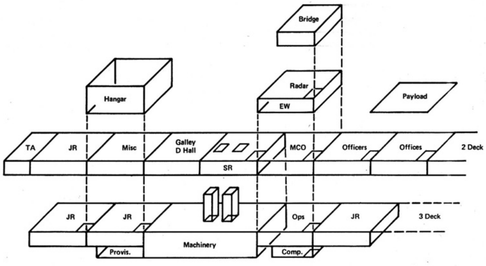


Fig. 4.8 – Early application of building blocks for frigate design (*Andrews [3]*)

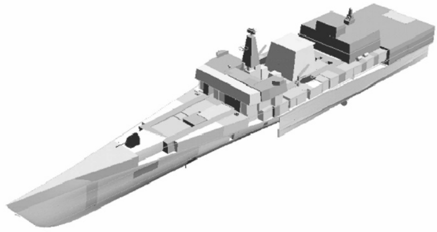


Fig. 4.9 – A Littoral Combat Ship comprising of building blocks (*Pawling [95]*)

and spatial descriptions in a single integrated approach to create ship designs that are 'naval architecturally balanced', i.e., feasible. Fourth, *Andrews [3]* handled the interaction between envelope and internal arrangement of systems by proposing to arrange blocks before 'wrapping' the envelope 'around' them; a topic discussed in Sections 4.4.2 and 4.5. Fifth, UCL emphasises human responsibility for design generation to create 'believable' results (*Andrews [4]*), at the cost of significantly reducing the number of alternatives considered (as discussed in Chapter 2).

4.3.2 NeVeSBu, RDM and TU Delft: SubCEM

An approach called SubCEM[1] was developed to support the early stage design of conventional, i.e., diesel-electric submarines (*Van der Nat [120]* and *Van der Nat et al. [121]*). It combines a knowledge-based shell named Quaestor (*Van Hees [124]*) with a space allocation routine called SUBSPACE to generate a numerical and spatial description of a submarine.

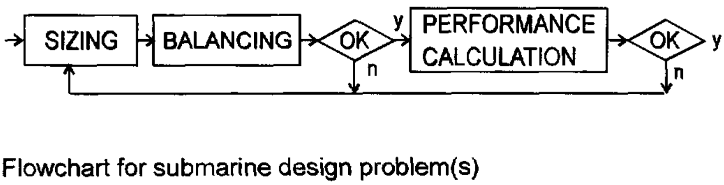


Fig. 4.10 – Overall process used by *Van der Nat [120]* to arrive at a weight and space balanced submarine: sizing, balancing and performance prediction

SubCEM uses a three-stage process to generate configurations (Fig. 4.10). First, it sizes spatial objects using mathematical rules from Quaestor's knowledge base. Sizing rules varied from simple look-up tables to complete models to determine electric motor size as a function of required power. Second, under human guidance, SUBSPACE packs three types of objects (discussed in Section 4.4.1) in a fixed sequence inside a 'cell' (Fig. 4.11). Aim is to establish whether the boat is sufficiently large to carry the systems required, not to arrive at an actual, detailed configuration. The size and allocation of objects is adjusted manually to arrive at a

[1] An abbreviation for: Submarine Concept Exploration Model

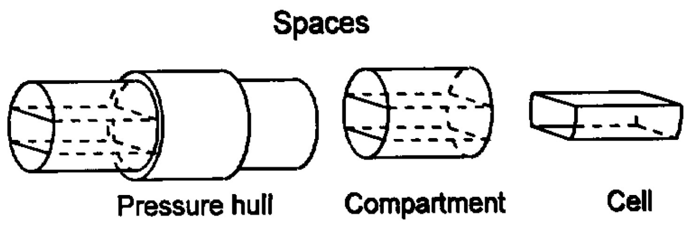


Fig. 4.11 – Decomposition of a submarine pressure hull into cells (Van der Nat [120])

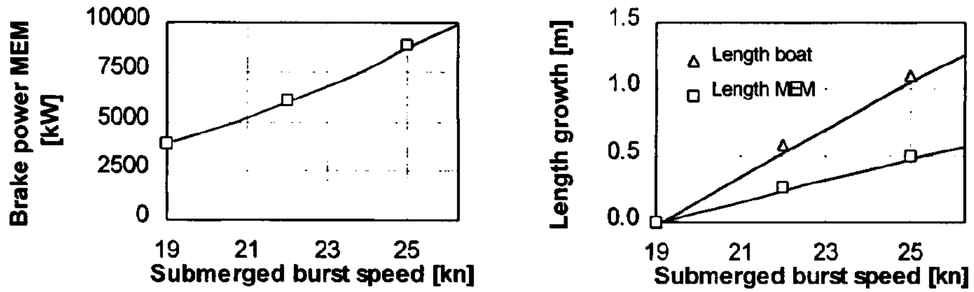


Fig. 4.12 – Sizing of boat length as a function of submerged speed (*Van der Nat [120]*)

'balanced' design that complies with the most important non-negotiable design requirements. Third, once all objects are packed, performance predictions assess, for instance, the boat's speed and endurance, using mathematical rules in the knowledge-base.

Van der Nat [120]'s approach has five relevant features for the packing approach developed in this Chapter. First, the approach in Fig. 4.10 inspired the overall work flow presented in Section 3.6 (notably Fig. 3.6). Main difference in Fig. 3.6 is balancing was merged with performance prediction, because the only difference between the two is the degree of negotiability of the design requirements.Second, *Van der Nat [120]*'s objects offer an alternative description of ship systems compared to UCL's Functional Building Blocks (Section 4.4.1 discusses their differences). Third, SUBSPACE offers a packing-based approach to generate a description of the submarine, i.e., objects have to adhere to rules regarding overlap, while other rules determine in which part of the boat objects can be packed (the 'cells' serve as positioning spaces). Fourth, the ship description is able to undergo automated changes (limited in scope: only a single variable to improve a single objective, using Quaestor's Newton-Raphson-based solver, but automated nonetheless) to meet performance requirements (see Fig. 4.12, for instance); an ability Section 4.1 argued to be essential. Fifth, SubCEM can concurrently change envelope size and internal arrangement while searching for 'balanced' designs that meet performance requirements (Fig. 4.12 shows the increase in boat length as a function of submerged speed), offering an alternative way to handle interaction between envelope and systems (see Sections 4.4.2 and 4.5).

4.3.3 University of Michigan

A research effort called ‘Intelligent Ship Arrangements’ was pursued at the University of Michigan at the same time as this author’s PhD research. Published in *Nick et al. [90]*, *Nick and Parsons [89]*, *Daniels and Parsons [28]* and *Nick [88]*, *Parsons et al. [94]* and *Daniels et al. [29]*, the effort focussed on arranging spaces, e.g., galley, storage rooms, fan rooms and crew accommodation, onboard a frigate. Arrangement takes place inside a fixed envelope subdivided by decks and bulkheads with fixed positions, i.e., ship size was determined prior to the arrangement process by a conceptual ship design model called ASSET (*Beyer et al. [16]*). Those parts of the ship that contain weapons and sensors, as well as propulsion and auxiliary systems were excluded from the arrangement process.

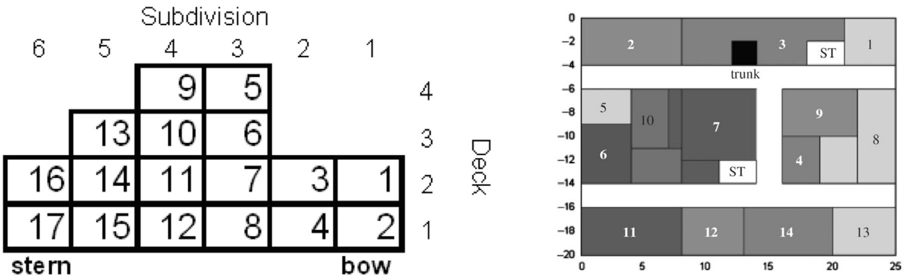


Fig. 4.13 – Decomposition of a ship into zone-decks (left) and spaces arranged on a zone-deck (right) (*Nick [88]*)

Packing of spaces in the ship is driven by a bespoke search algorithm and takes place in two steps. First, a space is allocated to a ‘zone-deck’, see Fig. 4.13 (similar to the ‘cells’ used by *Van der Nat [120]*). Second, the spaces inside a zone-deck are arranged into a coherent layout to ensure sufficient space is available and access to a -largely pre-defined- network of passageways and staircases is provided (Fig. 4.13).

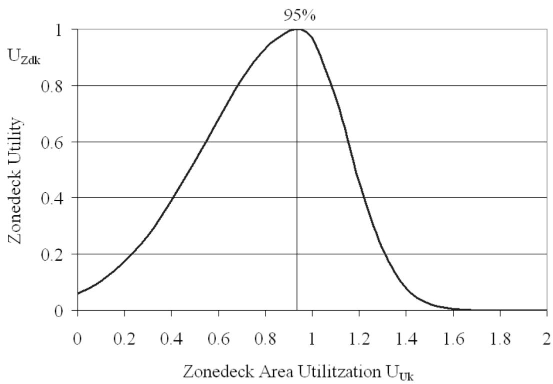


Fig. 4.14 – Fuzzy function expressing the preferred degree of area utilisation (*Nick [88]*)

The search algorithm uses so-called fuzzy objective functions (*Zadeh [141]*) to guide the search process. Such objectives use a pre-defined rating that captures the acceptability of a range of values for an objective function. Key feature of fuzzy functions are their normalised values which ranges between zero and one. Among other things, it enables the mathematical operations (such as addition and multiplication) on several fuzzy-functions to combine them into a single objective function. This enables the search algorithm to consider multiple ratings and reach a compromise between them based on pre-defined rules without human interaction. As such, it settles compromises in an a-prior way (Section 2.4.3). Note, the approach differs from the approach outlined in Section 3.5.2 to support compromise decisions. Considerable effort was spent formulating fuzzy functions based on a 'best-practice' manual on warship design from the US Navy (*Naval Sea Systems Command [84]*). The search effort stops if the objective fails to improve over a number of iterations. The design with the 'best rating' is considered to represent the 'best design', an approach discussed in Section 3.4.2, which neglects the implicit nature of compromise decisions.

Despite the large number of publications, it is still difficult to assess the utility of Michigan's approach in actual ship design projects. The reason is that its primary purpose is to support in-house ship design projects at the US Naval Sea Systems Command, the results of which have not been published so far.

Nonetheless, University of Michigan's approach has several relevant features the packing-based approach developed in this Chapter should consider or improve upon. Among the things to improve upon are the inability to vary envelope shape and size, the neglect of changes in centre of gravity that result from a different arrangement of spaces, the use of -largely- pre-defined passageways and staircases (though more recent efforts focussed on improving the flexibility of access routes, see *Daniels et al. [29]*), and the use of pre-defined fuzzy functions to determine compromises between negotiable design requirements in an a-priori manner, which affects acceptance of the results and limits the ability to identify and address flaws in ratings.

Still, one prominent feature stands out that the packing approach must replicate: within its restrictions, the 'Intelligent Ship Arrangements' approach is able to arrange spaces in a detailed three-dimensional ship description covering a major part of a warship.

4.3.4 Clemson University

The last packing approach to be discussed focusses on configuration design in mechanical engineering. The results were presented in *Grignon [53]*, *Grignon and Fadel [54]*, *Blouin et al. [17]*, *Miao [81]*, *Gantovnik et al. [45]* and *Miao et al. [82]*. Applications included the configuration design of a satellite (Fig. 4.15 and 4.16), a car engine, as well as a heavy transport vehicle (Figs. 4.17 and 4.18).

Clemson's approach has five relevant features for the packing approach developed in this Chapter. First, it can handle a wide range of mechanical systems, e.g., trucks and satellites, by packing free-form objects (defined by triangular meshes, e.g., Fig. 4.17) in an unbounded positioning space. Second, it considers the centre of gravity of the whole configuration during the search process, which is crucial for the generation of ship configurations with non-negotiable requirements on initial stability. Third, in the satellite application, the envelope (with fixed shape and size) was packed together with other objects (it is the conical object in the upper-right corner of Fig. 4.16). This approach offers yet another way to handle the interaction between envelope and the arrangement of systems inside (to be discussed in Sections 4.4.2 and 4.5). Fourth, the approach can generate configurations with a variable packing density (Fig. 4.15), e.g., to improve performances such as maintainability (Fig. 4.16), a topic Section 4.4.5

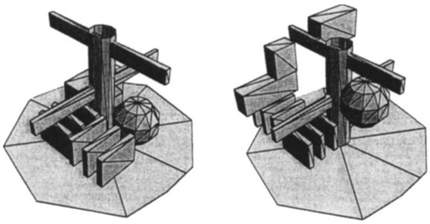


Fig. 4.15 – Compact (left) and spacious (right) satellite configurations (*Grignon [53]*)

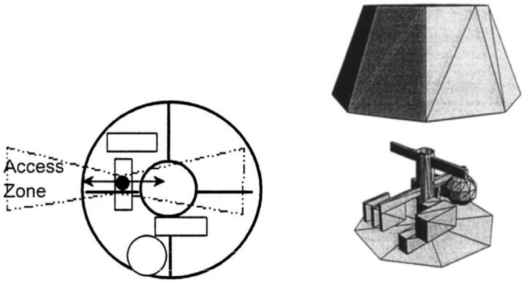


Fig. 4.16 – Access zone used to assess maintainability (*Grignon [53]*)

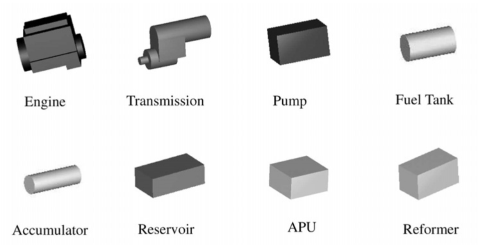


Fig. 4.17 – Truck components used by *Miao et al. [82]*

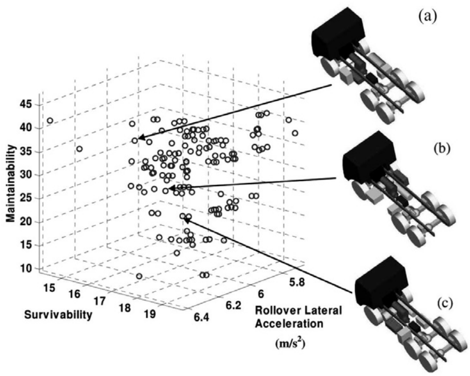


Fig. 4.18 – Plot of objective functions with corresponding truck designs (*Miao et al. [82]*)

elaborates upon. Fifth, the approach considers multiple objectives independently through the use of a 'Pareto-based' search algorithm (unlike *Nick [88]*), offering the user a range of alternatives to choose from after finishing the search process, as discussed in Section 3.5.3 (see Fig. 4.18 for an example).

4.3.5 Summary

The four packing approaches illustrate different ways to handle a packing problem, each tailored to a particular application. Some parts of the approaches will be used directly and without modification, other parts have to be developed from scratch, while yet others will be modified considerably to handle the configuration design of service vessels.

The resulting packing approach is developed in the remainder of this Chapter, a process that starts by discussing packing requisites in Section 4.4.

4.4 Packing requisites

A packing problem defined as having the following five requisites.

- Objects to be packed.
- A positioning space to pack the objects in.
- Rules governing overlap between objects.
- A way to detect overlap between objects.
- A way to enforce overlap rules.

They are discussed in detail in Sections 4.4.1 to 4.4.5, respectively.

4.4.1 Objects to be packed

What are objects?

Using packing as a parametric description of a service vessel first and foremost requires the proper definition of objects to be packed (an example of objects are the blocks in Fig. 4.5). It is these objects that will describe a service vessel. Their correct formulation is crucial for addressing the problems from Chapter 1 as the objects must be able to describe a wide range of systems, for example. In their most basic form, objects describe the spatial, numerical and other characteristics of a part of the ship.

Two references from Chapter 2 and Section 4.3: *Van der Nat [120]* and *Andrews and Pawling [8]* provide definitions of objects suited to handle the vast range of systems of a service vessel at a variable level of detail (Section 4.1). The other references, e.g., *Grignon [53]* and *Nick [88]* provide definitions of objects that handle a single level of detail and were not considered further in the development of objects to be packed.

Van der Nat [120] focusses on early stage submarine design. He uses a decomposition to break down a submarine into a collection of components. This results in *'hundreds of components'*, which is not *'efficient for conceptual design'*. Instead, they *'should be clustered to so-called 'objects'*. Fig. 4.19 shows the decomposition and clustering process. The objects that result from it are *'models predicting the properties of clustered groups of components'*. What components will be clustered, with what accuracy and what purpose depends on the design problem at hand, according to *Van der Nat [120]*.

Furthermore, *Van der Nat [120]* categorises three types of objects. Type I objects have discrete dimensions, i.e., length, width and height; Type II objects have a required surface area and height, while Type III objects are defined by volume only. The three types of objects are placed, by a dedicated packing algorithm: SUBSPACE, inside the pressure hull; Type III objects can also be placed in-between pressure hull and outer hull (discussed in Section 4.4.2).

Andrews and Pawling [8] use Design Building Blocks to describe a ship. *'The Design Building Block should be thought of as a placeholder or folder in the design space containing all information needed to describe a particular function. Building Blocks could have different*

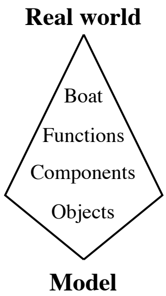


Fig. 4.19 – Derivation of objects by *Van der Nat [120]*

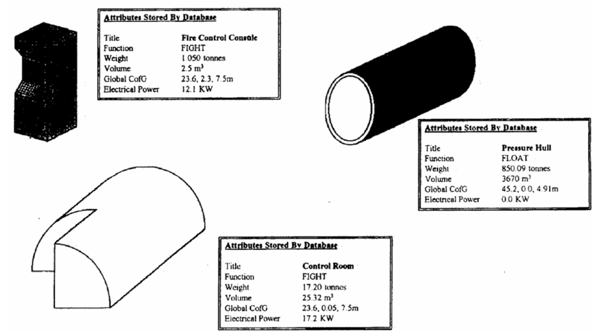


Fig. 4.20 – Examples of Design Building Blocks, from Andrews et al. [10]

combinations of properties; a geometric and numerical (weight) definition for a control console, or a weight and location definition only for a distributed system' according to *Pawling [95]* (based upon earlier work by *Andrews and Dicks [7]* and *Andrews and Pawling [8]*). Fig. 4.20 shows several examples of Design Building Blocks.

Both *Van der Nat [120]*'s objects and *Andrews and Pawling [8]*'s building blocks can describe the ship's geometrical, numerical and other characteristics, the most fundamental ability of a parametric ship description. Two differences stand out, however.

- First, *Van der Nat [120]* proposes clustering to be tailored to the design problem at hand. *Andrews and Pawling [8]*'s building blocks represent a function, i.e., a part of a ship with a single specific function. Though *Van der Nat [120]*'s approach uses functions as the basis for decomposing a submarine into components, it refrains from using function-based clustering.

  When considering the need to describe a ship at an appropriate but variable level of detail (Section 4.1), the function-based approach of *Andrews and Pawling [8]*'s building blocks is considered to be more restrictive[2]. For example, the drilling derrick of a drillship (see Chapter 6) can be clustered in a single object following *Van der Nat [120]*, even though it consists of many systems that fulfill multiple functions. Modelling the same drilling derrick would, according to their definition, require the use of several of *Andrews and Pawling [8]*'s building blocks.

- Second, *Van der Nat [120]*'s objects have a strict formulation (based on their ability to change shape) for use in a packing approach without direct human interaction. *Andrews and Pawling [8]*'s building blocks, on the other hand, have a generic geometrical description that can have any shape. This freedom is very useful in combination with human interaction, as it allows humans to invent shapes during the design process (Section 2.4).

  It also makes *Andrews and Pawling [8]*'s building blocks less suitable for a packing-based ship description used by a search algorithm operating without human interaction (Section 3.2). For such applications, the Type I, II and III objects proposed by *Van der*

[2]Though a more pragmatic application of the requirement to have a building block represent a function could largely resolve this.

Nat [120] are considered to be better suited as they do not need invention during the search process. *Van der Nat [120]*'s objects will therefore serve as the starting point for the development of the objects used in this dissertation.

These two differences cause the author to consider *Van der Nat [120]*'s objects to be more suitable for the packing approach developed in this Chapter than *Andrews and Pawling [8]*'s building blocks. One modification to *Van der Nat [120]*'s definition will be made: the term 'components' will be replaced by 'systems' to be consistent with the terminology in Section 1.2 and Appendix A.

The packing-based ship description will therefore use *objects defined as a collection of clustered systems* as the basis for the objects that will describe the ship. Note, an object can also represent a cluster with a single system.

The set of object types suitable to describe service vessels is discussed below.

A collection of object types suitable for service vessels

Van der Nat [120]'s approach requires objects to retain the relevant characteristics of the systems clustered in them (Fig. 4.19). This means that a set of object types suitable for service vessels should represent the relevant spatial characteristics of these systems. In this dissertation this concerns their function, shape and ability to change shape without loss of functionality.

Seven types of objects are proposed which together can describe the entire design of a service vessel (at a level of detail deemed suitable by the naval architect). Of the seven types, six object types represent physical parts of the ship, while the seventh is an artificial construct that offers the naval architect the means to constrain the configuration and focus it towards particular arrangements. Chapter 6 will use these object types to develop two parametric ship descriptions, respectively for a drillship and a frigate. Note, it is always possible to expand the range of objects to better describe particular systems onboard a ship, should the need arise, e.g., to better define the operating area of a crane as function of lifted load.

- Envelope object. The ship's envelope provides the carrying platform and sheltered space required for other systems (*Klein Woud and Stapersma [69]*). It consists of a water-tight, streamlined hull part (blue in Fig 4.21) and a superstructure part of a more arbitrary shape (pink in Fig 4.21).

  The hull's streamlining enables efficient motion through the water; shape changes to this part should not impair this, nor should they affect the water-tight integrity.

  The superstructure extends over the full width of the hull, as well as upwards from the damage control deck to a user-defined height (though its shape can be changed later on in the packing process). This provides space for systems located in the superstructure; e.g., sensors and weapons on a warship. Again envelope shape changes may not impair this function, e.g., a crane supported by a superstructure should not loose this support due to a change in superstructure shape.

- Subdivision objects. Service vessels have decks and bulkheads to subdivide the envelope into compartments. Like the superstructure, they extend over the full width of the envelope. Subdivision objects serve several purposes. Both decks and bulkheads provide structural integrity and enable containment and separation of systems onboard the ship. Moreover, decks increase internal surface area on which to carry other systems, while bulkheads contain flooding in case of damage (see Fig. 4.22 and *Van Diessen [122]*).

Decks and bulkheads are modelled respectively as horizontal and vertical planes that extend over part or whole of the envelope and superstructure. Watertight bulkheads cannot undergo shape changes that affect their ability to safeguard against flooding (e.g., an opening in a bulkhead below the damage control deck). Decks on the other hand can have openings without problems (with some exceptions, such as the weather deck).

- Hard objects. Hard objects represent systems that fulfill a wide range of functions. Common feature is that their arbitrary shape cannot change without loss of function of the cluster of systems they represent. As such, they resemble the Type I objects used by *Van der Nat [120].* Examples include a propulsion system (see Fig. 4.23), the drilling derrick on a drillship and weapons and sensors systems on a frigate.
- Soft object. Soft objects, like hard objects, represent systems that fulfill a wide range of functions. Key features are that these systems can be defined by either required surface area or volume (similar to the Type II and III objects used by *Van der Nat [120]*) and that their shape may change without loss of function. As such, their initial block-like shape is adjusted when required, whilst taking into account user-defined restrictions on their dimensions, e.g. minimum width. For example, the shape of the fuel tanks in Fig. 4.24 is adjusted to fit inside the ship's hull. Accommodation, storage, fuel tanks and vehicle decks can all be modelled as soft objects.
- Free space object. Some systems require additional space around them in order to work correctly. Free space objects enable one to model this additional space and provides the systems -when required- with an interface with the outside world. Free space objects have a fixed, block-like shape that can extend to the edge of the positioning space (see Fig. 4.55 for an example).

  Examples of systems that are modelled using free space objects include cranes and weapons. The former needs to be able to lift its load, transfer it and place it on another suitable position on or off the ship, all of which requires more space than necessary for the crane alone. The latter, e.g., a missile system on a frigate, need a firing arc free of obstructions to work correctly (the green object in Fig. 4.25).
- Connection objects. Service vessels have a multitude of networks of connections that ensure that the ship's systems work correctly. The connection object helps to model such connections. Connection objects are generated using rule-based routing performed by Dijkstra's algorithm (*Dijkstra [38]*), following the approach outlined in *Asmara and Nienhuis [12].* Connection objects require the presence of two other objects that form the start and end point of the connection and, therefore, cannot be packed independently.

  Their shape depends on the rules used during the routing; different rules generate different networks, however the use of a shortest-path algorithm means that connection objects prefer to have an 'as-straight-as-possible' shape. Examples include transport routes, access routes, up-and downtakes (Fig. 4.26) and ship services networks (which all use different rules for their individual definition).
- Logical objects. Section 4.1 argued the necessity of focussing the ship description. Part of this focussing can be achieved using so-called logical objects. Such objects enable the relative positions of the other types of objects to be constrained as required by

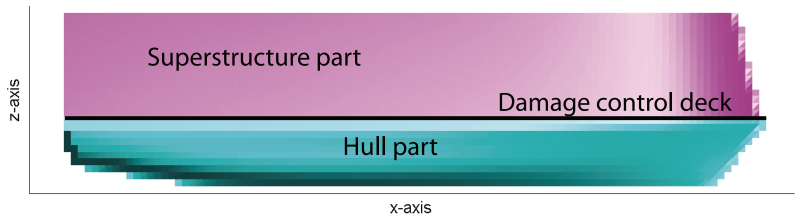


Fig. 4.21 – Envelope object

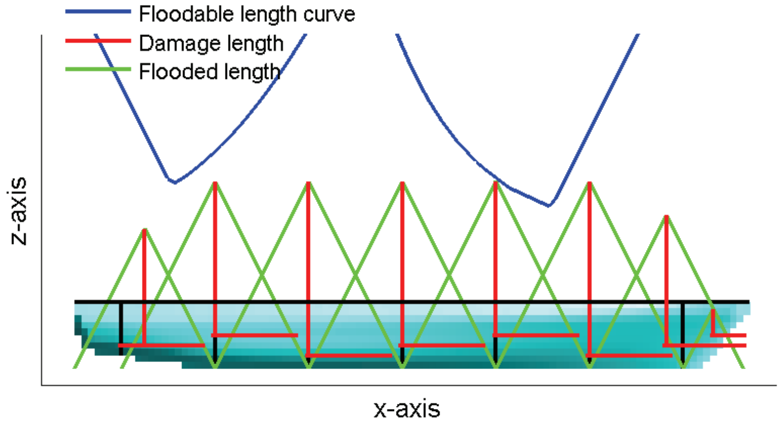


Fig. 4.22 – Subdivision objects: watertight bulkheads and the damage control deck (both in black)

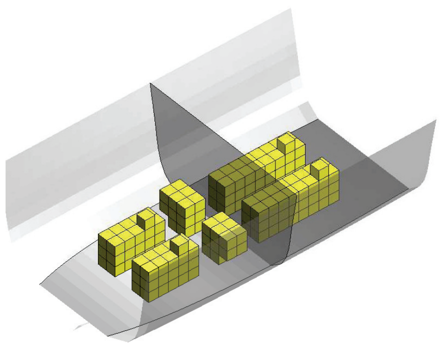


Fig. 4.23 – Hard objects representing a propulsion system

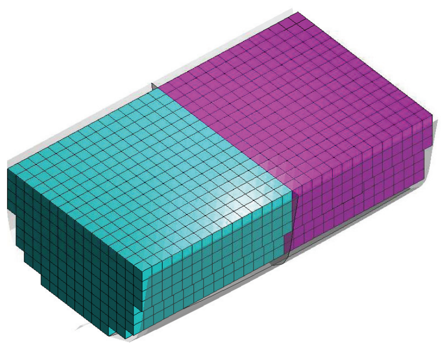


Fig. 4.24 – Two soft objects representing fuel tanks

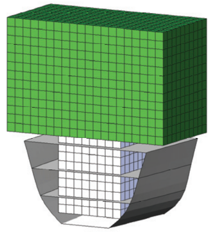


Fig. 4.25 – A free space object (firing arc of a missile system, in green)

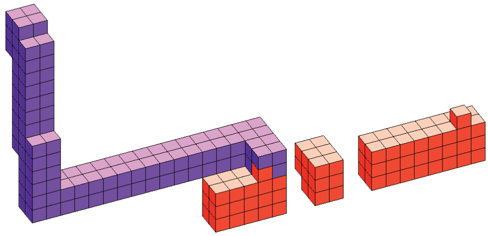


Fig. 4.26 – A connection object (uptake of a propulsion system)

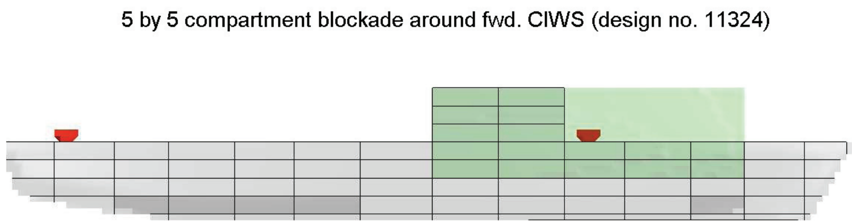


Fig. 4.27 – Five-by-five compartment blockage around a close-in-weapon-system ensure separation between redundant systems

the naval architect. For example, one can ensure separation of redundant systems by applying a logical object that blocks one of the redundant systems from occupying that part of the ship shared with the other system (as shown in Fig. 4.27 where a blockage ensures separation between two weapon systems on a frigate design from Chapter 6). As such, logical objects do not represent a cluster of systems with a physical presence. Instead, they are artificial objects are to be used in combination with the six object types discussed previously.

The seven types of objects form the basis of the parametric ship description. They enable the naval architect to **describe an entire service vessel at a variable level of detail**, the very first requirement outlined in Section 4.1 resulting from the discussion in Chapters 1 to 3. The way it does so is discussed below.

- Describing an entire vessel. For this, the naval architect must use the seven types of objects in combination. The envelope and subdivision objects are obviously required, whilst the remainder of the ship is described using a large number of hard, soft, free space and connection objects. To prevent unwanted variations, logical objects can help control the relative positions of the other objects.

  Both applications of the packing approach in Chapter 6 provide an overview of the objects used. Using the objects to describe a ship obviously results in a large number of objects, which the packing process must pack successfully. Section 4.5 will discuss how the entire set of objects will be packed.

Note that inventing the objects that represent the parts of the ship description involves considerable human effort (following the discussions in Section 1.6, 2.4.2, 3.5.1 and 3.6).

- Describing the ship at a variable level of detail. Both the decision to use *Van der Nat [120]*'s objects defined as bespoke clusters of systems as well as the definition of the seven distinct types helps the description to handle a variable level of detail.

  The former, the bespoke clustering, means the naval architect can add or remove systems from objects as required offering direct control over the level of detail.

  The latter, the seven object types, can use a low level of detail, e.g., describe the ship with a few, compartment-sized soft objects, or at a higher level of detail by using a large number of hard objects, free space objects and connection objects.

  The naval architect can -manually- select those types of objects deemed most appropriate. A variable level of detail also affects the way input parameters change object positions; a topic to be elaborated below.

Describing a service vessel is not sufficient. The description and therefore the objects it uses must be able to be changed by using input parameters to enable the application of search algorithms, which is discussed below.

Changing objects' properties using input parameters

Section 3.5.1 illustrated how input parameters could alter object characteristics, i.e., their presence, number, shape and size, position and connectivity. Of these characteristics, shape, size and position will be considered here for each object type. Presence, number and connectivity is only of interest when considering the entire set of objects used to describe service vessels, which are discussed in Chapter 6.

- Shape and size. An object's shape can be changed using input parameters in various ways. For example, the shape of a hull form can be altered using several continuous parameters that affect bulb shape (see Fig. 4.28). In addition, continuous parameters can alter hull dimensions: length, beam, draught and height (see Fig. 4.31). More complex shape changes can be achieved when one uses a discrete parameter to select different hull types from the library with existing objects (again, Fig. 4.31).

  Object shape and size can also depend on mathematical models called 'sizing models' by *Van der Nat [120]*. These range from a simple look-up table that determines gas turbine size as a function of required propulsion power to a complex, physics-based model to determine main electric motor size for a diesel-electric submarine (*Van der Nat [120]*).

  Still, most of the objects to be packed have a fixed shape and size; main exceptions are mobility related objects, e.g., propulsion plant, shafts, propulsors, uptakes and downtakes and fuel tanks.

- Position. Input parameters change the position of objects in the positioning space. The references from Chapter 2 used two types of parameters: combinatorial (e.g., by *Lee et al. [72]* and *Nick [88]*) and continuous (e.g., by *Grignon [53]* and *Grignon and Fadel [54]*). Both are discussed below.

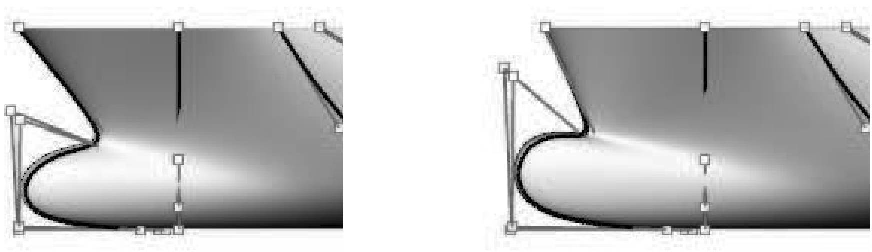


Fig. 4.28 – Continuous changes to a bow shape (from *Van Oers et al. [131]*, using the approach from *Goris [48]*)

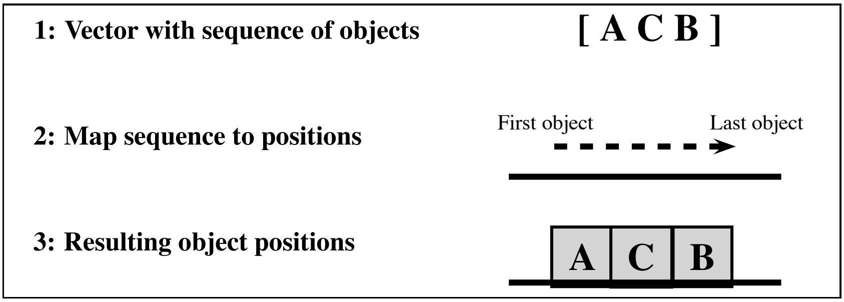


Fig. 4.29 – Positions defined by sequence and mapping

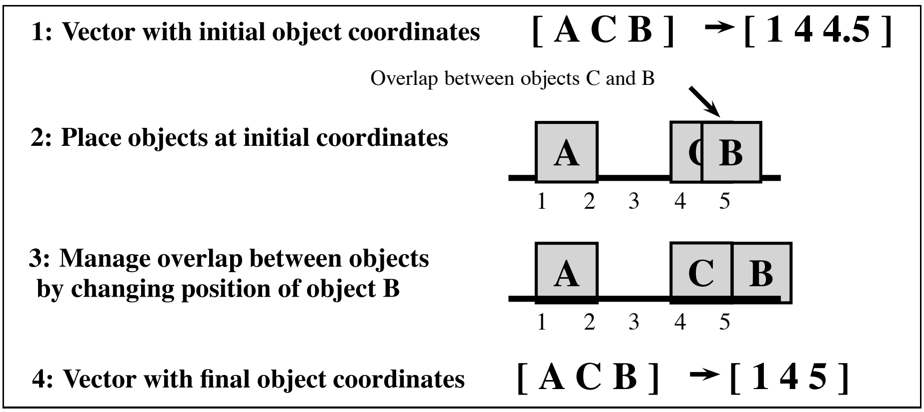


Fig. 4.30 – Positions defined by coordinates

- Combinatorial position parameters. Combinatorial parameters define the sequence in which objects A, B and C are packed (upper part of Fig. 4.29). Sequence is insufficient in itself and is translated into positions using a 'mapping' (middle part of Fig. 4.29). A pre-defined curve and a starting point determine where to place the first object A; objects C and B are packed adjacent to the earlier objects along the curve while preventing overlap (lower part of Fig. 4.29). Different sequences result in different positions. Key features are the constant and high packing density this achieves and rather coarse resolution with which objects can be placed.
- Continuous position parameters. Continuous parameters use coordinates in a user-defined coordinate system directly, instead of relying on sequence. The object's positions are defined by a vector (upper part of Fig. 4.30). They are placed at these positions as a second step (middle part of Fig. 4.30). Subsequently, overlap rules are used to check whether objects may overlap and, if not, the object positions are adjusted (lower part of Fig. 4.30).

The two types of position parameters differ in their ability to handle a wide range of service vessels. Closer consideration reveals continuous position parameters have several advantages over combinatorial position parameters.

First, continuous position parameters can handle space in-between objects, i.e., enable a variable packing density (measured by the volume of objects divided by the internal volume of the envelope). Service vessels use such space for several purposes.

- Weight-driven ships. Service vessels may be space-driven or weight-driven (*Watson [138]*). For space-driven ships, size is determined by the space needed to accommodate systems. The size of weight-driven ships follows from the buoyancy required to carry systems; as a result they have 'excess' space onboard. Designing weight-driven ships means one must be able to describe the ship including space-in-between objects, as it has a lower packing density than space-driven ships.
- Variable level of detail. The space in-between objects helps handle a variable level of detail (one of the requirements laid out in Section 4.1). Image two different levels of detail. The first uses objects representing compartments (which result from the envelope subdivided by decks and bulkheads) meaning the ship is **completely full** with objects and has a packing density of one. The second uses objects that represent systems inside compartments, i.e., has a higher level of detail, which means the ship is **partially full** and has a packing density lower than one. Handling a variable level of detail thus requires describing ships with a variable packing density, something continuous position parameters can achieve, and combinatorial position parameters cannot.
- Performance. Some of the performances of a service vessel, e.g., sea-keeping, may improve when increasing ship size. If this occurs whilst leaving other systems unchanged, the additional space inside the ship might lead to a lower packing density, including extra space in-between objects. Similarly, production cost may reduce if one builds less dense ships (e.g., see *Grant [51]*).

Second, continuous position parameters offer much more control over the resolution with which objects are placed. Combinatorial position parameters have a minimum change in position that is defined by the smallest object to be packed. Continuous position parameters on the other hand can place objects at a resolution deemed appropriate by a naval architect, enabling the search algorithm to place multiple objects such that their combined centre of gravity meets stringent non-negotiable requirements on stability and trim (discussed in Section 1.4).

Third, combinatorial position parameters rely on overlap rules that prevent all objects from overlapping with each other to translate sequence into position. Continuous position parameters offer more flexibility in this respect, allowing some objects to overlap while prescribing others not to overlap. Section 4.4.3 derives a bespoke set of overlap rules suitable for the early stage design of service vessels by exploiting the flexibility offered by continuous position parameters.

All three advantages lead to the decision that the packing approach in this dissertation will use **continuous position parameters, i.e., x, y and z-coordinates, to alter object positions**. This enables the packing approach to describe a wide range of service vessels (including weight-driven ships) at a variable level of detail, with the required accuracy to comply with non-negotiable requirements on trim and stability.

4.4.2 A positioning space

The positioning space that holds the objects forms a key feature of packing problems. Therefore, one must decide what to use as a position space in which to pack the objects defined in Section 4.4.1. The references from Chapter 2 provide three options. Each is discussed below.

1. *Smith et al. [110]*, *Van der Nat [120]*, *Lee et al. [72]* and *Nick [88]* all use the ship's envelope as a positioning space. *Smith et al. [110]*, *Lee et al. [72]* and *Nick [88]* left the envelope unchanged during the packing process, while *Van der Nat [120]*'s application in the field of submarine design allowed envelope length to change automatically; other aspects such as hull diameter remained constant.

2. *Andrews and Dicks [7]* and *Andrews and Pawling [8]* uses objects called 'design building blocks'. These are packed first, before an envelope is 'wrapped' around the set of objects. This sequence is intended to provide the naval architect with more freedom to pursue innovative configurations. *Andrews and Pawling [8]* mentions neither a positioning space nor packing explicitly. However, one can imagine a positioning space placed around the set of design building blocks large enough to enclose them all. (e.g., as in Fig. 4.2).

3. *Grignon [53]* and *Grignon and Fadel [54]* developed a packing approach for the configuration design of complex mechanical systems, i.e., satellites and car engines. *Grignon [53]* packs the envelope (of a satellite) simultaneously with other objects, instead of afterwards (as *Andrews and Pawling [8]*), in a large box-like positioning space meaning the envelope was not used as a positioning space; enclosure of objects was checked afterwards.

Section 4.1 listed the need to handle large changes to the entire ship description, including the shape and size of hull and superstructure. Such changes have a considerable effect on the configuration of systems in the service vessel, i.e., there is considerable interaction, which the packing process needs to cope with.

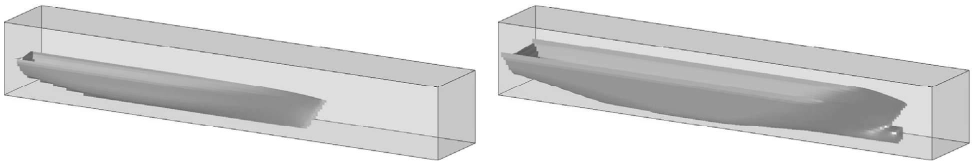


Fig. 4.31 – Two hulls with different shape and size packed in the same positioning space

The first option: using the envelope itself as a positioning space, will not offer sufficient flexibility in dealing with such interactions and is therefore discarded. Instead, the shared feature from the second and third option (*Andrews [3]* and *Grignon [53]*) is used: a box-like positioning space sufficiently big to cover the entire ship, including envelope (as in Fig. 4.31).

Using a box-like positioning space enables the envelope to be packed together with other objects (as in *Grignon [53]*). It provides the freedom to consider the different models for the interaction between envelope and system configurations for a wide range of hull shapes (see Fig. 4.31). Most importantly, changing the sequence of packing (discussed in Section 4.5) enables the envelope to be packed first: *Nick [88]*, last: *Andrews [3]* or somewhere in-between: *Grignon [53]*. Section 4.5 investigates the impact of different sequences on the range of configurations the packing-based ship description generates and pays particular attention to the interaction between envelope and the systems it carries (see Section 4.5.4).

One preliminary note regarding the use of the positioning space. Purpose of the positioning space is to hold the objects defined in Section 4.4.1 that together describe the ship. Since these objects will be packed using a packing process (to be discussed in Section 4.7), the positioning space will gradually fill up with objects (assuming not all objects are packed concurrently, see Section 4.5). As a result, the positioning space needs to be updated frequently to establish at any given instance whether space is available to pack a particular object.

4.4.3 Overlap rules

Overlap between objects was mentioned several times already in the previous Sections as a key feature of packing problems (Section 4.2). Overlap concerns the situation where the same space is occupied by two or more objects. Whether this is preferable depends on the type of overlap, i.e., whether overlap is **required**, **permissible** or **prohibited**. Required overlap means that objects must occupy the same space at the same time; prohibited overlap forbids this situation, while permissible speaks for itself.

Overlap rules define the overlap type between pairs of object types. As such, a single object type may overlap with the first object type but may, simultaneously, be prohibited from overlapping with the second object type. Overlap rules are the key to the generation of coherent ship configurations and must therefore be drawn up in a careful manner. Table 4.1 states the overlap types between the seven types of objects defined in Section 4.4.1. They are discussed in more detail below.

- Envelope object. Overlap between the envelope and most hard, all soft and all connection objects is required, as these objects represent systems located inside the ship's envelope. Subdivision is similar; decks and bulkheads are always considered to be fully enclosed by the envelope. All free space objects and those hard objects which are located outside the envelope are prohibited from overlapping with the envelope. An example of the former is the operating area of a crane; an example of the latter is the flight deck on a frigate. Those hard objects representing systems that protrude through the envelope have a permissible overlap, e.g., a large crane with its foundation may be located both inside and outside the envelope.

Overlap between two envelope objects is not considered, as only a single envelope object will be used. The reason is that the envelope encloses a single, closed and connected volume, regardless of whether it is a mono-hull, SWATH[3] or semi-submersible (the latter is used for floating oil rigs). Note, such a closed solid may very well need to be constructed from multiple surfaces.

- Subdivision objects. Subdivision objects: decks and bulkheads, may overlap with other decks and bulkheads in order to subdivide the envelope into smaller compartments. Both hard and soft objects may either overlap with bulkheads, or not. The former occurs when objects represent large systems that extend over a large part of the ship, e.g., a large accommodation area on a drillship. The latter occurs when objects are to be located in-between decks and bulkheads. Overlap with free space objects is prohibited, as this interferes with the latter object's ability to provide an unobstructed space such as the firing arc of a gun. Connections on the other hand can overlap with subdivision objects, e.g., to represent uptakes and passageways penetrating decks and bulkheads.

- Hard objects. Hard objects have the simplest overlap rules of all; they cannot overlap with other hard, soft, free space and connection objects (the envelope and subdivision objects were discussed already). The reason is straightforward. Hard objects represent physical systems that cannot occupy the same space occupied by soft, free space or connection objects.

- Soft objects. Soft objects are not allowed to overlap with other soft objects, free space objects and connection objects for reasons similar to those of hard objects. Note that soft objects may alter their shape to prevent overlap (see Section 4.4.5).

- Free space objects. Free space objects may overlap with other free space objects. The reason is simple, the free space around systems may typically overlap, e.g., the firing arc of a gun may overlap with the field of view of the ship's bridge. Logical objects offer the opportunity to prevent such overlap in situations where it is to be avoided. In addition, free space objects can either have permissible or prohibited overlap with connection objects. The former is relevant for a transport route to move a helicopter from flight deck to hangar. The latter occurs when uptakes obstruct the operating area of a crane.

- Connection objects. Connection objects may not overlap with other connection objects to prevent unwanted interference between, say, an uptake and a transport route.

Overlap rules for logical objects have not been discussed, so far. They are discussed below.

- Logical objects. Logical objects enable the naval architect to focus the ship's configuration when and where appropriate. To do so, they rely on flexible, user-defined overlap rules that define overlap not between types of objects but between individual objects in

[3]SWATH stands for 'small water-plane twin hull'. It is a catamaran type hull shape.

the ship description. The logical object is subsequently either prohibited or required to overlap with these objects, while overlap with all other objects is permissible (i.e., their presence is ignored). This ability can be used, for example, by combining a hard object representing a radar mast in combination with a logical object that is required to overlap with a bulkhead. This provides the radar with structural support. Section 4.6 discusses logical objects and other focussing methods.

The overlap rules in Table 4.1 are meaningless without the ability to detect unwanted overlap situations and the ability to address them. For the former, one needs a geometry definition (discussed in Section 4.4.4). For the latter, one needs design changes for overlap management (discussed in Section 4.4.5).

Object type	Envelope	Subdivision	Hard	Soft	Free space	Connection	Logical
Envelope	- - -	required	required permissible prohibited	required	prohibited	required	required permissible prohibited
Subdivision	- - -	permissible	prohibited permissible	prohibited permissible	prohibited	permissible	required permissible prohibited
Hard	- - -	- - -	prohibited	prohibited	prohibited	prohibited	required permissible prohibited
Soft	- - -	- - -	- - -	prohibited	prohibited	prohibited	required permissible prohibited
Free space	- - -	- - -	- - -	- - -	permissible	prohibited permissible	required permissible prohibited
Connection	- - -	- - -	- - -	- - -	- - -	prohibited	required permissible prohibited
Logical	- - -	- - -	- - -	- - -	- - -	- - -	required permissible prohibited

Table 4.1 – Overlap types between pairs of objects

4.4.4 Geometry definition: the key to overlap detection

The rules governing overlap between objects require the ability to detect overlap, which requires a mathematical description of the objects' geometry and positions. The latter were discussed already in Section 4.4.1. The former uses a mathematical geometry definition and the selection of a definition greatly influence the computational effort required to solve the packing problem. An overview of suitable definitions is discussed in *Cagan et al. [25]* and *Blouin et al. [17]*. Broadly speaking two different definitions are used; each is discussed below.

- Surface definition. Surface definitions describe space an object occupies by defining its outer boundary. Surface definitions come in various forms. *Grignon [53]*, *Grignon and Fadel [54]* and *Miao et al. [82]* uses triangular meshes. Alternatively, one can use a spline-based definition such as NURBS (e.g., *Piegl and Tiller [99]*), available in the CAD-program *Rhinoceros*, *McNeel and Associates [80]*).

  Overlap can be detected by calculating the intersection between the surfaces of multiple objects. Main benefits include the ease with which complex shapes can be described and the wide availability of computational routine to perform intersection operations. Drawbacks include the inability to check for overlap when one object is located inside another object (e.g., as is the case when one object is required to overlap completely with another), see Table 4.1, and the relatively slow speed of intersection calculations.

- Solid definitions. Solid definitions describe the solid enclosed by an object's outer boundary, e.g., consider enclosed volume. They again come in a range of different types, e.g., constructive solid geometry (*Dykstra and Muuss [40]*) and voxels. Voxels are volume elements that subdivide the positioning space into smaller, repetitive element (voxels stands for volume pixel, an analogue with the smallest discrete unit of a computer display). *Nick [88]* uses block-like voxels, i.e., a grid, laid out in the ship's envelope, while *Jacquenot [62]* uses sphere-based voxels to model free-form objects.

  Solid definitions enable a more thorough overlap detection and have the ability to detect overlap when one object is required to overlap completely with another object. Moreover, proper use of a solid definition enables the packing process to search not for overlap, but for available space, making the overlap management process more efficient. Some solid definitions, e.g., block-like voxels as used by *Nick [88]*, also help when routing connection objects, as the voxels can readily be covered into input for routing (the adjacency between voxels can be covered into a so-called graph which routing algorithms use, such as in *Dijkstra [38]*).

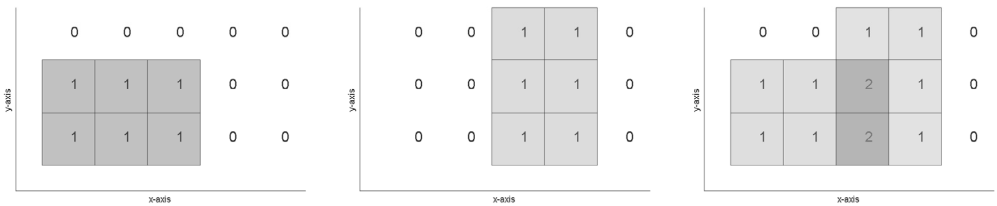


Fig. 4.32 – Two objects defined by voxels (left, middle) and overlap detection through matrix addition (right)

The packing approach in this dissertation uses a solid definition: the block-shaped voxels used by *Nick [88]*. Reasons are threefold.

- First, such voxels can detect overlap in situations where one object overlaps completely with other objects. This is important because of the use of the 'required overlap' rule in Table 4.1.
- Second, voxels can be represented in a large matrix, enabling the use of computationally efficient matrix calculations offered by *Matlab* (*The Mathworks Inc. [114]*) to perform overlap detection and removal. Fig. 4.32 offers a simple example. Shape and position of two objects are represented in two matrices by voxels with a value of 1 (left and middle images in Fig. 4.32). By adding them together, the overlapping voxels can be identified by their value of 2, a simple and rapid method of overlap detection (right image in Fig. 4.32).
- Third, block-like voxels offer support for the routing approach using *Dijkstra [38]*'s algorithm which is necessary to define connection objects.

The surface-based representation will not be discarded altogether, as the envelope object will initialled be described using a surface-based definition (the envelope width for every x, z position). The reasons are two-fold. First, hydrostatic calculations on block-like voxels are too inaccurate and second, the envelope surface will be used for visualisation purposes (Fig. 4.31).

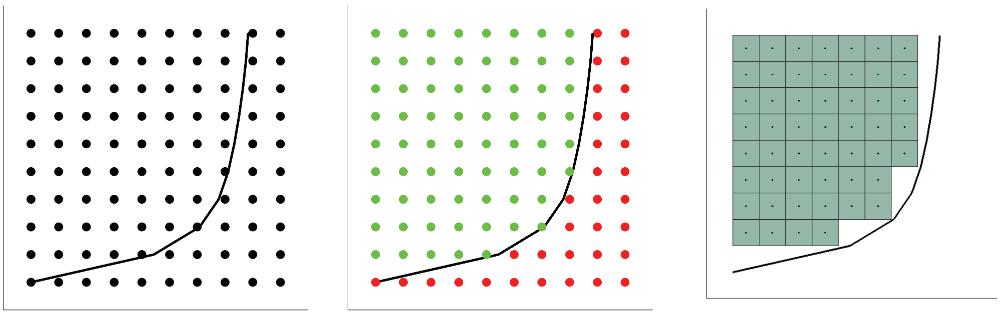


Fig. 4.33 – Generation of voxels from envelope cross-section: from voxel corner points and envelope cross-section (left), through interior and exterior voxel corner points (middle), to internal voxels (right)

Using a surface-based envelope definition in combination with a voxel-based solid definition means one must have a method to derive voxels from surfaces. This process is shown in Fig. 4.33. First, one defines the voxel corner points in *Matlab* (left image in Fig. 4.33). Next, one uses an inequality operation to establish which points lie inside and outside the envelope (middle image in Fig. 4.33). Lastly, one combines the corner points into voxels that describe the enclosed volume inside the envelope (right image in Fig. 4.33), similar to those used to define other objects (such as the hard objects in Fig. 4.23 or the soft objects Fig. 4.24).

The block-like voxels used in the packing approach currently have a fixed dimension of 1 by 1 by 1 m. Even though the packing approach handles resolutions other than this unit cube with ease, the implementation used to generate the applications in Chapter 6 did not (due to some short-cuts taken early on in the development of the packing implementation). Still, further development of this packing implementation will be able to handle variable voxel size with ease. Furthermore, some voxels may be filled partially, i.e., to use volume enclosed in voxels located partly inside and partly outside the envelope. This increases the accuracy of volume calculations without increasing the number of voxels required.

4.4.5 Design changes for overlap management

Rules for overlap as those proposed in Table 4.1 help assess whether the shapes and positions of objects in the positioning space are acceptable. Still, merely checking whether overlap rules are complied with is insufficient, however, because one must also be able to change the objects' position and / or shape to ensure that unwanted overlap situations are addressed. Defining a set of design changes that can do this ensures the packing approach can generate coherent ship configurations, i.e., configurations that comply with Table 4.1.

Three design changes are proposed to this end: two shape changes and a position change. The design changes transform the **initial shape and position** of an object into a **final shape and position** that does comply with the overlap rules. They are discussed below, using two simple objects that are prohibited to overlap as an example (shown in Fig. 4.34 to 4.39).

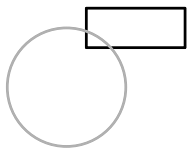


Fig. 4.34 – Prohibited overlap between circle and rectangle

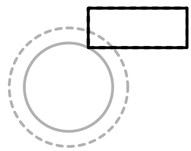


Fig. 4.35 – Removed overlap by changing radius of circle

- **Change shape using an object's own shape function**. First option to remove overlap is changing the shape of the circle, e.g., by changing the circles' radius, Figs. 4.34 and 4.35. This change relies on a change in parameter value used by the object's own shape function (changing multiple parameter values concurrently is also possible). The shape function ensures a coherent object shape after overlap management, preventing unwanted shape alterations, e.g., changes that affect the streamlined nature of hull part of the envelope object. Note that the resulting shape is that particular shape that deviates the least from the initial shape, while still meeting all overlap rules. Hence, aspects such as enclosed area or volume of an object are not considered during overlap management.
- **Changing shape by conforming an object's shape to the shape of another object**. An alternative shape change for overlap management sees part of the shape of one object being used by another object. For example, in Figs. 4.36 and 4.37, the circular object's shape is adjusted such that the overlapping part is removed. As a result, part of the circular object's shape conforms to the shape of the box-like object. This shape change, the rectangular gap in the circular object's shape, can result in shapes that cannot be created by altering parameter values of the function that defines the circular object's initial shape.

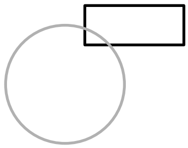


Fig. 4.36 – Prohibited overlap between circle and rectangle

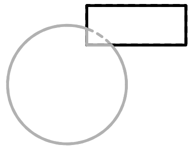


Fig. 4.37 – Removed overlap by conforming circle's shape to that of the rectangle

- **Change position of one of the objects**. This is the design change most commonly used by the packing references discussed earlier. In the example in Figs. 4.38 and 4.39, the position of the rectangular object is changed by moving it to the right, thereby removing the prohibited overlap. It places objects on an initial position determined by the search algorithm; overlap management subsequently shifts the object to the position closest to the initial position that complies with all overlap rules in Table 4.1.

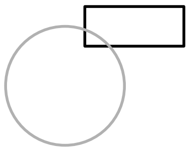


Fig. 4.38 – Prohibited overlap between circle and rectangle

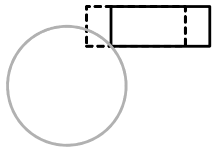


Fig. 4.39 – Removed overlap by changing position of rectangle

The three kinds of design changes listed above are used by the seven object types discussed in Section 4.4.1, as shown in Table 4.2. Note, not all objects use all design changes. For instance, the envelope is the only object that may use the first type of shape change, while all other objects use position changes. This is considered further in Section 4.5.5.

Design changes	Change position	Change through shape function	Conform own shape	Change envelope shape
Envelope	not relevant	yes	yes	not relevant
Subdivision	yes	no	yes	no
Hard	yes	no	no	no
Soft	yes	no	yes	no
Free space	yes	no	no	yes
Connection	not relevant	no	yes	no
Logical object	yes	no	yes	yes

Table 4.2 – Design changes per object type (see Table 4.10 for the updated version)

Table 4.2 also shows that a fourth design change is added: objects may alter the envelope's shape using either of the two shape changes. Dedicated changes to an individual object are, strictly speaking, a sequentiality issue that Section 4.5 should address.

Nonetheless, the reason it is discussed here is that adjusting envelope shape enables free space objects to adjust the topside part of the envelope to create the required field of view (e.g., for a radar system). Without this ability, the interaction between envelope and free space objects would be insufficiently flexible. Such inflexibility could force one to adjust the envelope shape prior to the packing process to accommodate particular systems, which immediately freezes the position of these systems during the packing process. For example, if the envelope shape is adjusted to provide the required free space around a helicopter flight deck, the position of the helicopter flight deck itself (e.g., modelled using a hard object) cannot change position anymore, which would leave little freedom to vary flight deck configuration. These latter changes to the envelope shape may have unintended consequences, as Section 4.7 explains.

Examples of how each of the four design changes are used by the object types are discussed below. They show two situations. First, the situation before applying design changes to manage overlap, i.e., presenting the initial shape and position of an object and, second, the situation after overlap management, i.e., after applying the design changes from Table 4.2. The discussion also provides an overview of the typical use of input parameters to alter an object's initial shape and position. Logical objects are discussed separately, in Section 4.6.

Envelope object

Fig. 4.21 showed the envelope object consists of a hull part and a topside part. Each shape change in Table 4.2 is applied to hull and superstructure, both individually and in combination. Note that these changes may not interfere with structural support, enclosure or the watertight integrity of the envelope, or with objects that have been packed already.

Parameter no.	Changes	Type
1	Hull type	Discrete
2	Length over all	Continuous
3	Maximum beam	Continuous
4	Design draught	Continuous
5	Stretching of bow	Continuous
6	Stretching of stern	Continuous

Table 4.3 – Parameter-driven changes to envelope shape and size

- **Initial shape and position**. The initial shape of the envelope is based on a pre-defined parent hull shape with a superstructure extending upwards to the upper boundary of the positioning space (Fig. 4.41). Six parameters change hull shape and size (see Table 4.3). Hull shape can undergo both shape changes using a discrete parameter (as shown earlier in Fig. 4.31 and 4.28) and by using two continuous parameters that alter the fullness of bow and stern (change the prismatic coefficients of bow and stern). In combination, they can stretch the parallel mid-ship and alter the location of the centre of buoyancy (Fig. 4.40). Envelope length, width and design draught can also change using three continuous parameters (again, shown earlier in Fig. 4.28). The envelope's initial position is fixed. The stern is placed on the aft boundary of the positioning space (see Fig. 4.31).

- **Change shape using an object's own shape function**. The envelope's own shape function is the same function used to determine its initial shape and position. As such, one can adjust the parameters (e.g., width) to enclose objects that must overlap with the envelope. Fig. 4.42 shows an example where the envelope's width is adjusted to ensure a helicopter hangar of a warship is fully enclosed. Such a change in width retains the coherent nature of the envelope shape and provides compliance with the overlap rules.
- **Conform shape**. Shape conformation is used by the superstructure to adjust its shape to that of other objects. Three types of shape conformations are used. First, the shape conforms itself by removing those parts of the superstructure that overlap with free space objects (Fig. 4.43). Second, the shape is conformed to provide support for objects located on top of the envelope (Fig. 4.44) to prevent them from floating around in thin air. Third, the shape is conformed to enclose those objects to be located inside the envelope (Fig. 4.44). Shape changes as those in Fig. 4.43 may not interfere with the watertight integrity of the hull and, hence, are prohibited below the damage control deck.
- **Change position**. The envelope's position does not change during the packing process and has a fixed location in the positioning shape. The reason is simple, a position change would only change the envelope position relative to the position of other objects. For simplicity's sake, these other objects will change position instead (Table 4.2).

Fig. 4.45 shows the envelope object after the shape changes (compare to Fig. 4.41). Parts of the superstructure that do not enclose or support objects are removed to get this final shape.

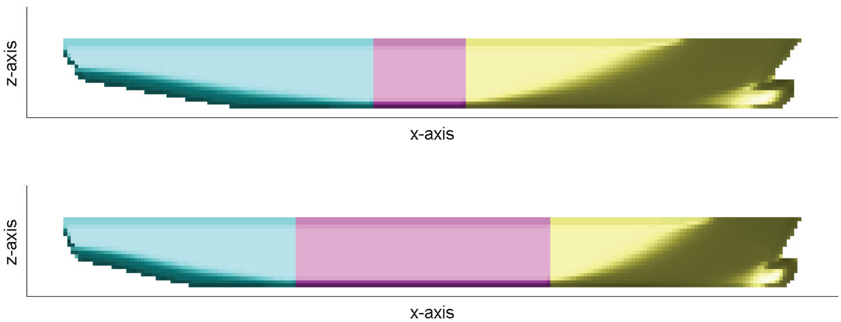


Fig. 4.40 – Length variation of the parallel mid-ship

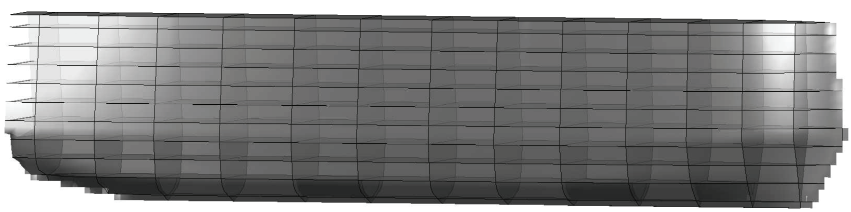


Fig. 4.41 – Initial envelope shape

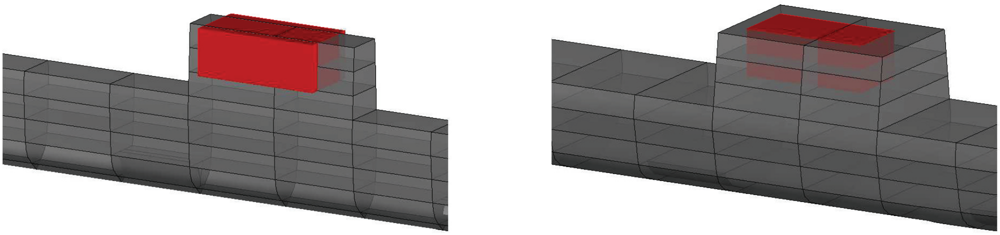


Fig. 4.42 – Enclosing a hangar by increasing envelope width

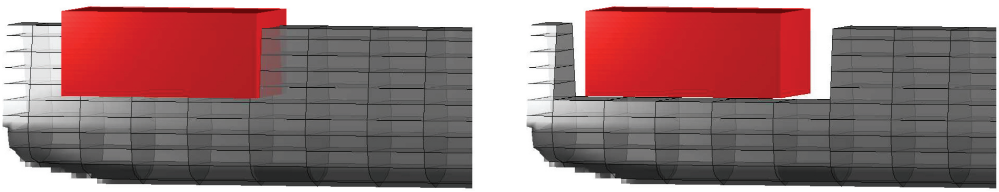


Fig. 4.43 – Conforming the envelope shape to remove overlap with a free space object

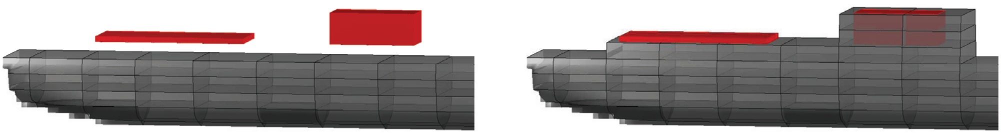


Fig. 4.44 – Conforming envelope shape to ensure enclosure and structural support

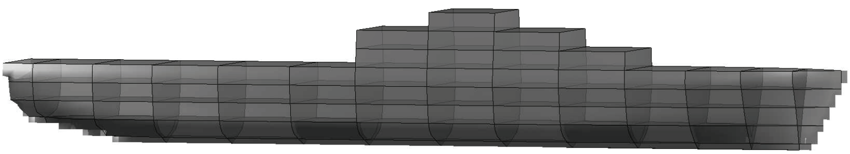


Fig. 4.45 – Envelope after shape conformation

Subdivision objects

Subdivision objects fall in one of two types: decks and transverse bulkheads, that together subdivide the envelope into smaller compartments. Longitudinal bulkheads may also be defined, using the modelling discussed below. Decks and bulkheads differ with regard to the design changes they undergo.

- **Initial shape and position**. The initial shape of decks and bulkheads is simple. Decks are modelled as horizontal planes that extend over the full width and length of the positioning space and have a unique vertical position; bulkheads are vertical planes that extend over the full height and width and have a unique longitudinal position. Fig. 4.46 shows the initial shape and position of decks and bulkheads. The initial position of decks and bulkheads is determined by input parameters from the search algorithm (shown in Tables 4.4 and 4.5).

- **Conform shape**. Decks and bulkheads can conform their shape to that of other objects. For example, Fig. 4.43 shows how decks and bulkheads -in addition to the envelope- conform their shape to remove overlap with the red free space object. A similar effect is shown in Fig. 4.44 where the subdivision objects also conform shape to provide structural support and enclosure. Other examples include the generation of holes in decks or bulkheads. For implementation purposes, the generation of holes has been modelled by allowing subdivision objects to overlap with other object types (as shown in Table 4.1). Note that shape changes, e.g. holes, may not interfere with the watertight integrity of the ship and are therefore prohibited below the damage control deck.

- **Change position**. Subdivision objects may also change position. This has been implemented only for bulkheads, though the process is applicable for decks as well. Bulkhead positions influence the ship's ability to contain flooding when damaged and may not have openings below the damage control deck (for warships at least). This means that overlap of bulkheads with other objects cannot be addressed by shape conformation, but requires position changes instead.

  However, bulkhead position changes must take into account the bulkheads' role in containing flooding. To this end, the changes in bulkhead position are constrained using a floodable length curve to ensure bulkhead configurations can cope with the non-negotiable reserve buoyancy requirement at the design draught. The parameters in Table 4.5 provide the initial bulkhead positions (upper image in Fig. 4.48), positions that are adjusted to comply with the overlap rules in Table 4.1 and the reserve buoyancy requirement (lower image in Fig. 4.48). Key feature is that different input parameters yield different bulkhead configurations (the configuration should only meet overlap rules and reserve buoyancy requirements, i.e., the number of bulkheads is not minimised). Full details are provided in *Van Diessen [122]*.

Fig. 4.47 shows subdivision objects after altering position and shape (compare to Fig. 4.46).

No.	Changes	Type
1	Height of tank top	Continuous
2	Deck height	Continuous
3	Height of the damage control deck	Integer

Table 4.4 – Parameter-driven changes to decks

No.	Changes	Type
1	Number of bulkheads	Integer
2	Minimum bulkhead distance	Continuous
3	Reduction in bulkhead spacing at the bow	Continuous
4	Reduction in bulkhead spacing at the stern	Continuous

Table 4.5 – Parameter-driven changes to bulkheads

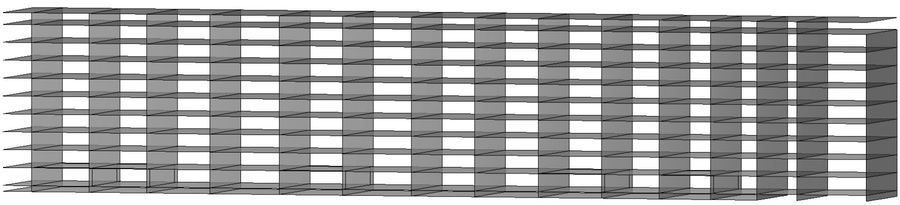


Fig. 4.46 – Initial shape of decks and bulkheads

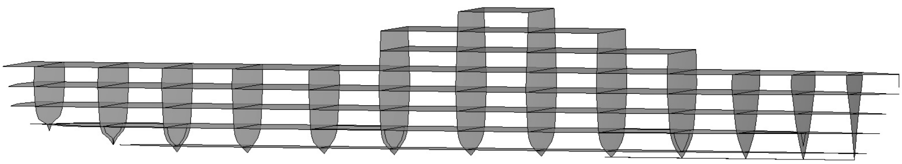


Fig. 4.47 – Shape of decks and bulkheads after shape and position changes

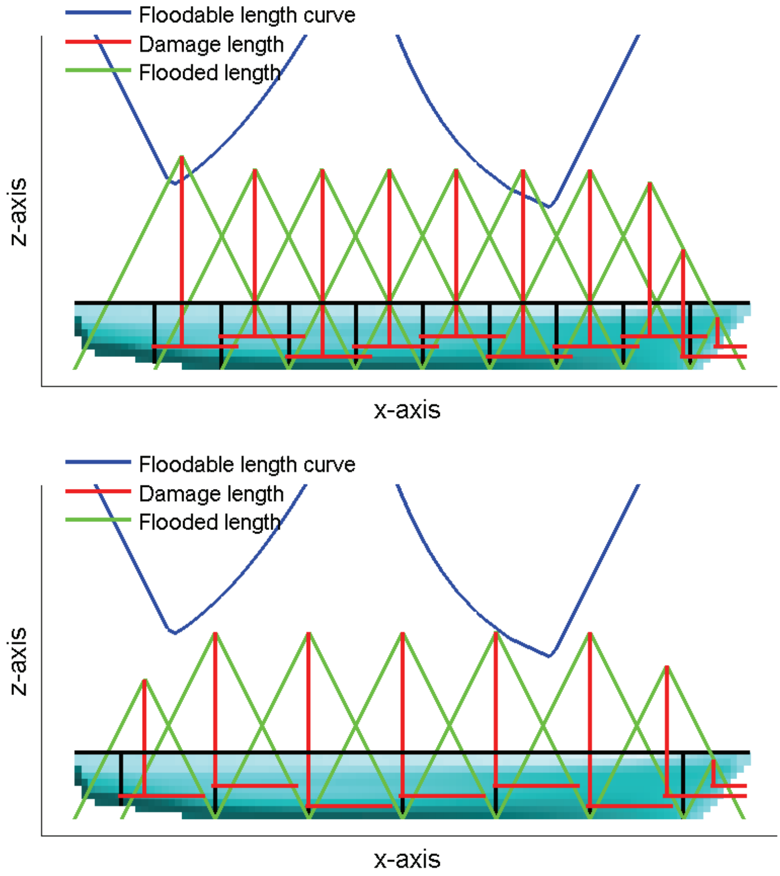


Fig. 4.48 – Change in bulkhead positions to comply with reserve buoyancy requirements (upper image: insufficient buoyancy, lower image: sufficient buoyancy)

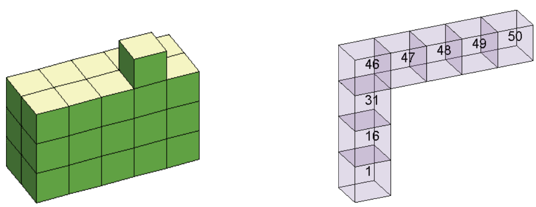


Fig. 4.49 – A diesel engine (31 voxels) and a simple crane (8 voxels)

Hard objects

Section 4.4.1 introduced hard objects. Their main feature is their inability to change shape without loss of the function of the systems they represent. This means that of the three design changes for overlap management discussed previously, only position changes remain.

- **Initial shape and position**. Hard objects can have any shape and, like other objects, are represented by voxels. These voxels are defined by referring to the indices of the voxels in the positioning space. Fig. 4.49 shows two examples, a diesel engine and a crane. If necessary, one can use a pre-defined list of objects to enable, for instance, the selection of a diesel engine on the basis of required power (if need be using an input parameter, see Table 4.6). The initial position of hard objects is determined by input parameters, see Table 4.6, following the discussion in Section 4.4.1. Note that the position parameters are integer parameters, as the use of voxels only allow finite positioning accuracy. Moreover, the number of input parameters can be altered as required. For instance, an object representing a drilling derrick on a drillship may be restricted to the ship's centreline, and thus requires only two parameters to change its initial x and z coordinates.

No.	Changes	Type
1	x-position	Integer
2	y-position	Integer
3	z-position	Integer
4	Parameter to determine shape and / or size	Continuous

Table 4.6 – Parameter-driven changes to hard objects

- **Change position**. Hard objects change their position to comply with overlap rules. For example, the weapon system in Fig. 4.51 must at least overlap partly with the envelope. To achieve this, it changes its initial position (left image in Fig. 4.51) to a position that does comply with the overlap rules closest to its initial position (the right image in Fig. 4.51).

  The following approach is used for position changes, assuming an initial position at x_i, y_i, z_i (also see Fig. 4.50). First, all x-positions are considered keeping the current y_i, z_i positions constant. Should a position be available, the object is placed at the x position closed to the initial x_i position. If no position is available, the y-position is altered and the search for a suitable x-position repeated. When all y-positions have been exhausted, z-position is changed and the process repeated for all x and y-positions. The result is that the entire positioning space is considered to position the hard object, with the final position being the closest to the initial position x_i, y_i, z_i.

  Moreover, the use of an initial and final position means overlap management is used only when required, prevent unnecessary use of overlap detection (which reduces the computational effort).

```
--------------------------- Preparation ----------------------------------

- Retrieve at initial x,y and z-position of the current system of
  objects as defined by search algorithm.
- Create vectors with available x, y and z positions to consider for this
  particular system during the position changing process
- Sort vectors with available x, y and z positions according to the
  distance to the system's initial position (from low to high distance)

----------------- Conditions to stop position changes----------------------

- Perform overlap detection and change positions using process below until
    - All objects in system can be packed
    - All available combinations of x,y and z-positions have been exhausted

----------------- Positioning changing process ----------------------------

    For counter_z = 1:length(vector_with_z_positions)
        current_z_position = vector_with_z_positions(counter)
        For counter_y = 1:length(vector_with_y_positions)
            current_y_position = vector_with_y_positions(counter)
            For counter_x = 1:length(vector_with_x_positions)
                current_x_position = vector_with_x_positions(counter)
                For object = 1:number_of_objects_in_sytem
                    Create voxels of current object
                    Perform overlap detection using voxels at position
                    (current_x_position, current_y_position, current_z_position)
                Next

                If All objects in system can be packed
                    Stop position changes
                    Store voxels occupied by each object
                    Flag = "system could be packed"
                Else
                    Flag = "system could not be packed"
                    Continue
                End
            Next
        Next
    Next

If Flag = "system could not be packed"
    Stop packing process, i.e., position changes did not result in compliance
    with overlap rules
Else If Flag = "system could  be packed"
    Store voxels for objects in current system
    Continue packing process with next system of objects
End
```

Fig. 4.50 – Pseudo-code outlining the approach used for position changes of hard, soft and free space objects

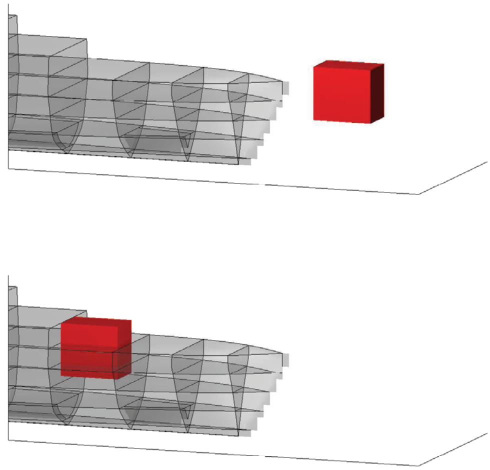


Fig. 4.51 – A weapon system achieves the required overlap by changing from the initial position (left) to the final position (right)

Soft objects

Soft objects differ from hard objects by their ability to alter shape while maintaining functionality (Section 4.4.1). Only shape conformation and position changes will be used for overlap management (Table 4.2); the other shape change: 'change shape using an object's own shape function' is not considered.

- **Initial shape and position**. Soft objects are defined by required surface area with a required height, or by required volume (following *Van der Nat [120]*). Many shapes comply with these requirements and a more stringent definition of the soft object's initial shape is necessary. A very simple initial shape is used: a box. The width and height of the box are user-defined, its length as of yet undetermined. Fig. 4.52 shows the initial shape of a volume-based soft object. Table 4.7 shows the input parameters a soft object may use.

- **Conform shape**. Soft objects may conform their shape to meet the overlap rules from Table 4.1. Such shape changes are subject to several constraints, however. First and foremost, the soft object must have the required surface area or volume. This is achieved by using numerical integration that determines the required length of the box. Second comes the minimum required width (for area-based soft objects) or area (for volume-based soft objects) in transverse direction, which controls the degree of shape conformation. Setting this minimum value equal to the width and height of the box will result in a rectangular box with the length required to achieve the required volume. Obviously, lower values result in a more flexible shape (compare both soft objects in Fig. 4.54 with different shapes but similar volume). Furthermore, the soft object needs to be connected in longitudinal direction to prevent it from being split into two or more objects. Similar checks are made to ensure transverse connectivity, to remove holes and to achieve the required height of soft objects. The naval architect can activate these checks when necessary.

- **Change position**. The soft object will change position if it cannot comply with the constraints on shape conformation discussed above. These position changes are similar to those of hard objects discussed previously. Fig. 4.54 shows an example, the increase in minimum value results in a shorter, 'fatter' soft object (compare right image with left image). This shorter soft object can only be located inside the envelope at a position further forward compared to the leaner soft object (compare right image with left image). Note that shape conformation and position changes are carried out concurrently.

No.	Changes	Type
1	Choice between area or volume	Discrete
2	x-position	Integer
3	y-position	Integer
4	z-position	Integer
5	Required surface area or volume	Continuous
6	Minimum transverse value	Continuous

Table 4.7 – Parameter-driven changes to soft objects

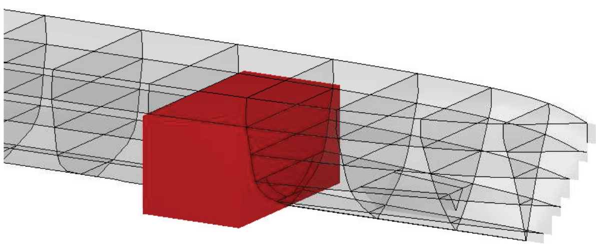


Fig. 4.52 – Initial shape of a soft object (using final length)

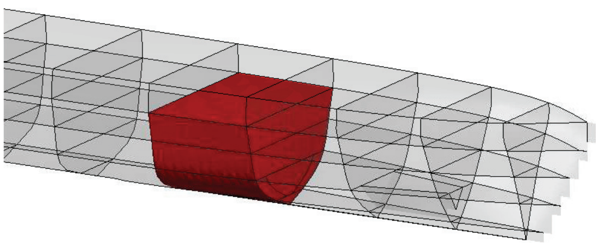


Fig. 4.53 – Final shape of a soft object conformed to the envelope

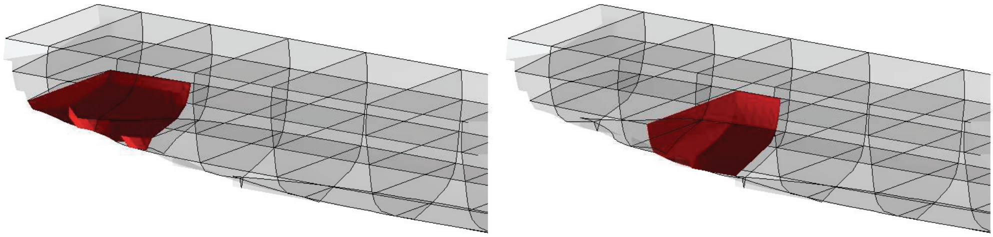


Fig. 4.54 – Adjusting the degree of shape conformation of a soft object (V = 300 m^3, minimum value per x-position is 0.01 m^2 (left) and 60 m^2 (right)

Free space objects

Free space objects represent the free space around particular systems to ensure they working correctly. For example, free objects may provide a clear operating area for cranes, sensors, weapons, helicopter platforms and davits. Free space objects use two design changes for overlap management: changes in position and changes to the envelope object (Table 4.2).

- **Initial shape and position**. The initial shape of a free space object is simple: a rectangular box (Fig. 4.56). Moreover, this initial shape does not change during the packing process (Table 4.2). Note that in reality, some required free spaces will have a distinct angular shape, e.g., a gun with a horizontal firing arc of $135°$. Still, such shapes are not modelled directly. Instead, they will be approximated by using several block-shaped free space objects concurrently, which helps to reduce the number of object types required.

  Key feature of the free space object is its size, which can be finite (like other objects), or more importantly, infinite (by extending to a particular boundary of the positioning space). This enables a single free space object, for example, to mimic the viewing arc of a radar system on a frigate. The naval architect defines in which of the six directions ($x-$, $x+$, $y-$, $y+$, $z-$ and $z+$) the free space object extends to the relevant edge of the positioning space (see Table 4.8). For example, the free space object shown in Fig. 4.25 extends to the upper edge and port and starboard side edges of the positioning space. The initial position of a free space object is determined in a similar manner as those of hard and soft objects (Table 4.8).

- **Conform envelope shape**. The way free space objects conform the envelope shape was shown and discussed already using Fig. 4.43. In this shape change, the prohibited overlap is removed by removing, i.e., 'chopping', that part of the envelope that overlaps with the free space object. Such envelope shape changes are based on the location of decks and bulkheads, to prevent structural discontinuities.

  Note, one might consider the use of free space objects to model large internal areas such as a vehicle deck. Unfortunately, the current packing rules discussed in this Section prohibit this. The part of the envelope enclosing the free space object will be removed using the envelope shape change described above, which will cause a large hole in this ship running from side-to-side. If desirable, modification of the design changes used by free space objects could address this unwanted behaviour.

- **Change position**. Free space objects first attempt to change envelope shape to manage overlap. Still, such shape changes may not affect the enclosure or structural support of objects already present in the ship description. Moreover, overlap with hard and soft objects cannot be managed by changing the envelope.

  In both cases, overlap is managed by changing position. The approach to shape changes are similar to that of hard and soft objects. Fig. 4.55 shows an example. The upper image shows the envelope with a superstructure shape that cannot be changed. The middle image shows the superstructure shape with the overlapping free space object at its initial position. The lower image shows the free space object at its final position, which does not overlap with the envelope.

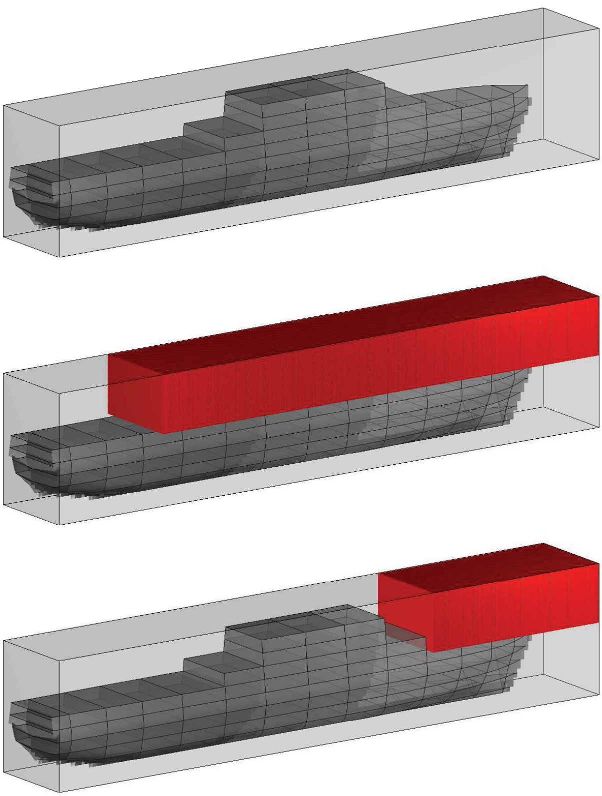


Fig. 4.55 – Envelope with fixed superstructure shape (above), initial position with overlap between free space object and envelope (middle) and final position after moving the free space object forward (below)

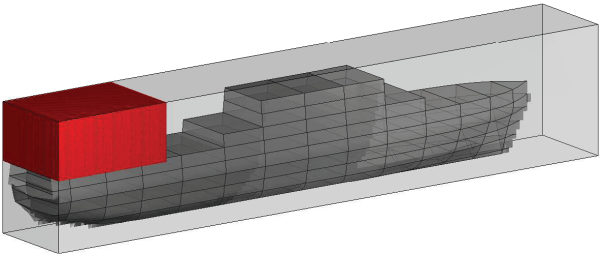


Fig. 4.56 – Shape of a free space object (red)

No.	Changes	Type
1	x-position	Integer
2	y-position	Integer
3	z-position	Integer
4	Length of the free space object	Continuous
5	Width of the free space object	Continuous
6	Height of the free space object	Continuous

Table 4.8 – Parameter-driven changes to free space objects

Connection objects

Connection objects have a special role: they connect two or more other objects. As such, they do not have the ability to change position on their own. Instead, they derive their position from the other objects they connect. Connection objects are generated by converting the voxels in the positioning space into a graph that indicates connectivity between voxels (see Fig. 4.57). Using this graph, one can exploit existing routing approaches such as proposed in *Asmara and Nienhuis [12]* and *Asmara and Nienhuis [13]* that are based on *Dijkstra [38]*'s algorithm to identify the shortest path between to objects to be connected.

Different rules ensure the generation of different types of networks, e.g., a transport route with an elevator or a network of passageways (Fig. 4.58).

- **Initial shape and position**. The initial shape of a connection object is simple, a connection that is as straight as possible. Should a straight connection be impossible, it will make the required connection with square angles (right image in Fig. 4.59). User-defined preference directions change the order of directions used (Table 4.9), for example, whether the horizonal distance is covered first, before traversing the vertical distance (left image in Fig. 4.59), or the opposite (right image in Fig. 4.59). The initial position of connection objects is determined by the two or more other objects they connect, which are defined using the first two parameters in Table 4.9.

- **Conform shape**. The initial straight shape of a connection object is conformed when other objects are present with which the connection object cannot overlap. An example is shown in Fig. 4.60. The left image shows the initial, i.e., straight, shape of a connection object. The right image shows the final shape of the connection object, after conforming its to several hard objects to prevent overlap.

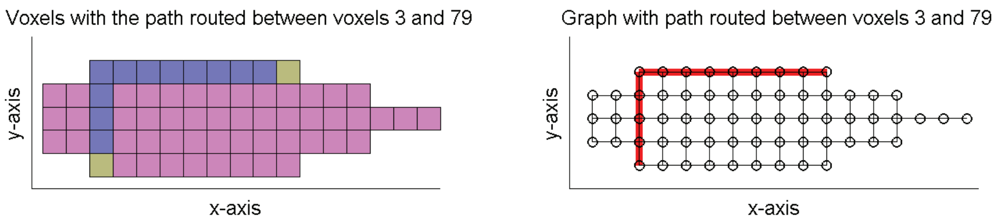


Fig. 4.57 – Connection object represented as voxels (left) and a path in a graph (right)

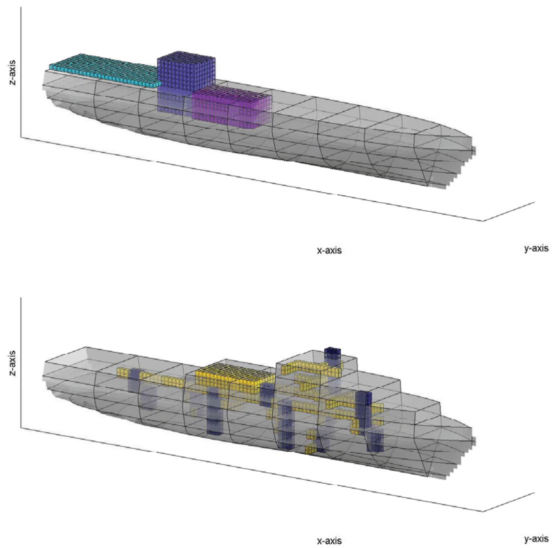


Fig. 4.58 – Examples of connection objects: transport route with an elevator connecting hangar with flight deck (top), network of passageways and staircases (bottom)

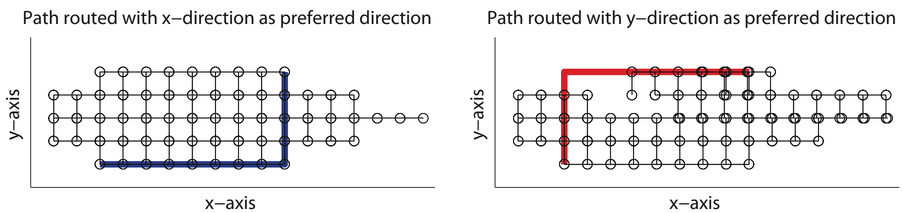
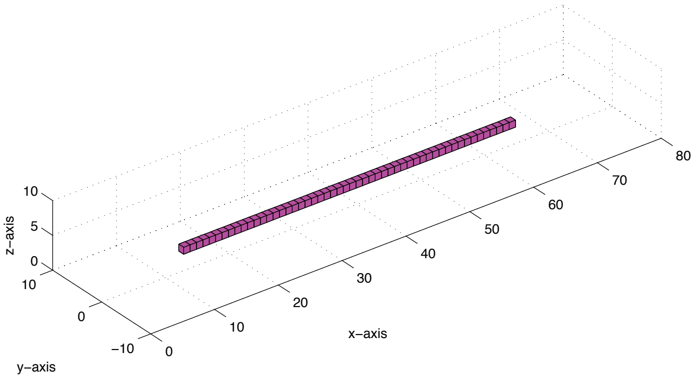


Fig. 4.59 – Different shapes resulting from different preference directions

No.	Changes	Type
1	First object to be connected	Integer
2	Second object to be connected	Integer
3	Preference for x-direction	Integer
4	Preference for y-direction	Integer
5	Preference for z-direction	Integer

Table 4.9 – Parameter-driven changes to connection objects



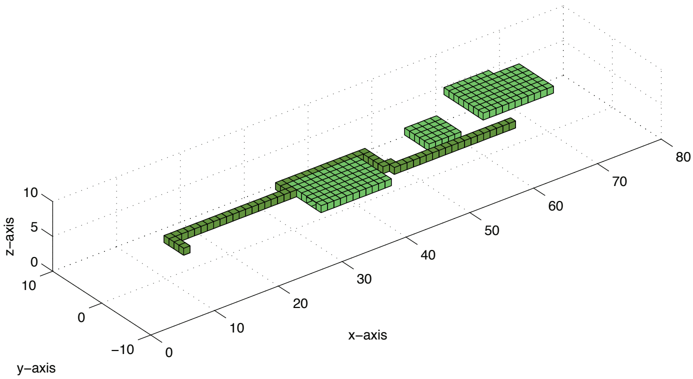


Fig. 4.60 – Initial shape (upper image) and final shape (lower image) of a connection object

Logical objects

Logical objects derive their shape and position from other objects they constrain. This means they share the design changes of these objects. For example, a logical object that accompanies a soft object will use the same kind of design changes as the soft object for overlap management.

Their overlap rules were discussed previously in Section 4.4.3 and can differ per individual logical object. If the overlap rules for the logical object are not met, it should lead to changes to the shape and / or position of the objects they constraint. How this is achieved is discussed in Sections 4.6 and 4.7.

Closure: design changes for overlap management

Section 4.4.5 discusses the three types of design changes applied to the seven objects (see Table 4.2). All design changes use the same three steps.

- First, definition of an object's initial shape and initial position.
- Second, detection of unwanted overlap of the object with other objects.
- Third, application of design changes to adjust the initial shape and / or initial position into the final shape and position.

The design changes enable the packing approach to manage overlap during the packing process to ensure that overlap rules from Table 4.1 are met for each and every object, and ,as such, ensure a coherent ship configuration.

4.4.6 Closure: packing requisites

This Section discussed the requisites needed to develop a packing-based approach able to generate service vessel configurations based on human input and changes in input parameters.

To this end, it discussed the seven object types used and the positioning space in which they are packed. It also covered -in detail- the crucial overlap rules and the design changes used to adjust the configuration to make it meet overlap rules. The latter, the design changes, convert the initial shapes and positions of objects (based on human input and input parameter values) with unwanted overlap, into the final configuration with adjusted object shapes and positions that complies with overlap rules. Of special importance is the parameterisation of positions. By using ordinary x, y, z-coordinates instead of combinatorial coordinates (the latter were used by *Nick [88]*), the packing approach can generate designs with a variable packing density, enabling it to cope with a variable level of detail, the generation of weight-driven and space-driven designs and, lastly, ships that are more spacious to improve performance, e.g., seakeeping or vulnerability.

Section 4.7 integrates the individual packing requisites into a coherent packing process that the search algorithm can use to generate coherent service vessel configurations.

4.5 Packing sequence

A prominent feature of the packing process is the sequence with which objects are to be packed. Sequence is necessary for a simple reason. The individual design changes discussed in Section 4.4.5 can all serve to manage overlap, but when one has two or more objects to consider, one must decide when to apply which design change. For example, if one has one hard and one soft object that overlap, one must decide whether to change the hard object, the soft object, or both. The same issue rose previously in Section 4.4.2 when dealing with the interaction between the envelope and other objects (‘wrapping’ according to *Andrews [3]* or ‘packing’ according to *Nick [88]*).

4.5.1 Applying design changes: concurrently, sequentially or a mixture?

The sequence of application of design changes lies within two extremes. Either one can apply the design changes from Section 4.4.5 sequentially, i.e., to a single object at a time, or one can apply the design changes from Section 4.4.5 concurrently, i.e., to all objects at the same time.

A mixture of concurrent and sequential packing is used by the packing approach developed in this dissertation, for the reasons outlined below.

- In general, a sequential approach will be used, which means that objects are to be packed ‘one by one’. Should unwanted overlap arise, the design changes from Section 4.4.5

are applied only to the object currently being packed, all prior objects are considered frozen and left unchanged in a manner similar to *Grignon [53]*. This simplifies the application of design changes considerably, as the shape and / or position changes have to be considered for a single object at a time.

- Still, truly packing objects 'one-by-one' is not always possible, for two reasons.

  – First, modelling systems may require the use of several different object types from Section 4.4.1. For example, using a single hard object to model a missile launcher on a warship (as in Fig. 4.25) is questionable, unless it is accompanied by a free space object that provides it with a clear arc of fire.

  – Second, different systems clustered in an object may use different sizing rules. In that case, it is much more convenient to model them in separate objects to enable their independent sizing. For example, one must use two different sets of sizing rules for a combined diesel-and-gas turbine propulsion plant. One set determines the size of diesel engines, the other set determines the properties of the gas turbines to be used.

  Both cases can be handled using a limited degree of concurrency. I.e., one must use several objects and pack them concurrently, i.e., in **systems of objects**. Examples of such systems of objects include a propulsion plant with its uptakes, a helicopter flight deck with its flight path and an accommodation area with a connection to the network of passageways and staircases.

  Note that relative positions of objects in a system of objects are fixed during the packing process, while the overlap rules from Table 4.1 are suspended in order to simplify modelling. The latter could, for example, allow overlap between a hard object (representing a weapon) and a free space object (representing the firing arc), which would be prohibited according to Table 4.1.

The result is that the packing approach presented in this Chapter packs a large number of systems of objects in a sequential manner. During the packing process, the objects in the system are packed concurrently while overlap among objects within the system is neglected.

This compromise between concurrency and sequentiality raises a new issue: first, what constitutes a proper sequence, a topic which includes the issue of when should one pack the envelope.

4.5.2 Establishing a proper packing sequence

The choice to use a predominantly sequential packing process means one must define an appropriate sequence in which the objects are packed. When defining such a sequence, one must consider that such a sequence is necessary to arrive at a workable application of the design changes from Section 4.4.5, making it important, but not an end in itself.

Instead, the consequences of particular sequences are more important, three of which were considered in the development of an appropriate sequence.

1. Whether a change in sequence affects the range of configurations the packing approach can generate.

2. Whether a change in sequence influences the yield and computational effort of the packing process.

3. Whether a change in sequence facilitates the ease of implementation, i.e., the programming effort required to make a particular sequence work.

The three consequences are each considered in detail, starting with the first issue.

4.5.3 Does sequence affect the range of configurations?

The first issue to be considered is the most important one: does sequence actually matter? At first sight, answering it appears to need a particular set of objects that together describe a particular service vessel. This, however, would make it difficult to draw generic conclusions.

Fortunately, one can draw generic conclusions from simpler examples, as the sequence affects the application of the three types of design changes discussed in Section 4.4.5. These design changes are shared by the seven object types from Section 4.4.1. Therefore, it will suffice to study only the impact of sequence on the application of the three design changes.

Starting point is the core of overlap management: the translation of the initial shape and position into the final shape and position discussed in Section 4.4.5. It is this translation and, most importantly whether it is necessary, that helps resolve the issue of whether sequence matters. The reason is that **sequence is relevant only when overlap management is necessary**, i.e., when design changes are applied to comply with the overlap rules. If overlap management is unnecessary, sequence of packing can be varied without effect on the outcome of the packing process because the initial shapes and positions become the final shapes and positions.

This can be taken one step further. Section 4.4.5 illustrated how the shape and position of an object prior to overlap management can be changed by input parameters (e.g., as in Table 4.2). If this variation of initial shape and initial position can generate the shape and position an object takes after overlap management it becomes possible to generate such shapes without requiring overlap management, in which case, the changes in input parameters nullify the effect of sequence.

Whether variation of the initial shape and position of an object is sufficient to remove the effects of sequence, depends on the number, bounds and types of input parameter values and, most importantly, on the design changes used by the object to manage overlap. The effect of design changes is discussed below and covers each of the three design changes introduced in Fig. 4.34 to 4.39.

1. **Change shape using an object's own shape function**. The first design change discussed in Section 4.4.5 alters an object's shape by adjusting values of parameters a mathematical function uses to determine its shape. Its initial shape is determined by the user-defined function (invented by humans) and the initial parameters values. Its final shape uses the same mathematical function, with different parameter values. This means that the shape before and after overlap management can be the same, provided the parameters can attain the same values (by sharing the same bounds). As this is generally the case, it means that the effects for sequence for a change shape using an object's own shape function will not impact the range of shapes the object may take.

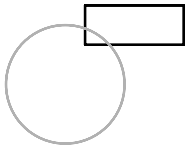


Fig. 4.61 – Prohibited overlap between circle and rectangle

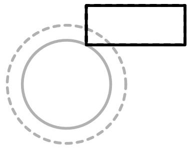


Fig. 4.62 – Removed overlap by changing radius of circle

A simple example is shown in Fig. 4.61 and 4.62 (repeated from Fig. 4.34 and 4.35). The circular and rectangular objects overlap (Fig. 4.61), which is removed by altering the circular object's shape (Fig. 4.62).

Assume the circular object uses a shape function with a single parameter: its radius, to determine its shape, and that this parameter is also an input parameter used by the search algorithm. With appropriate bounds, it is then possible to generate an initial shape of the circular similar to the final shape shown in Fig. 4.62), but without relying on overlap management. This means that the final shape of the circular object can be generated without considering when the circular object is packed, i.e., without impact of a change in sequence.

Obviously, a single example where sequence has no impact is not mathematical proof. However, it serves to illustrate that when shape changes are purely driven by changes in parameter values, it should not matter how these values are determined (via overlap management with an impact of sequence, or without overlap management and no impact of sequence), as only their values will be of relevance.

2. **Changing shape by conforming an object's shape to the shape of another object**. The second shape change conforms an object's initial shape to that of another object to remove prohibited overlap. As such, its final shape will be a combination of this own initial shape and of the other object's shape.

   This means that its final shape will likely to differ markedly from its initial shape (which is block-like for soft objects, see Fig. 4.52 and Table 4.7), making it is very difficult to change the initial shape such that it coincides with its final shape. As a result, the effects of sequence cannot be removed, because sequence determines whether and how an object's shape is conformed.

   A simple example using Fig. 4.63 and 4.64 helps to illustrate this. Again, assume the circular object uses a shape function with a single parameter: its radius, to determine its shape, and that this parameter is also an input parameter used by the search algorithm.

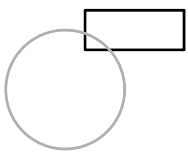


Fig. 4.63 – Prohibited overlap between circle and rectangle

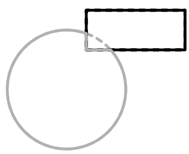


Fig. 4.64 – Removed overlap by conforming circle's shape to that of the rectangle

Overlap is managed by conforming the circle's shape to that of the rectangle. The final shape, a circle with a 'bite' taken out of it, differs considerably from the original shape. More specifically, it is a shape that cannot be created using the object's own shape function, which can describe only circles. Instead, its new shape function uses parameters from both the rectangular and circular object.

This means that sequence does become important, because packing the circular object first and adjusting the rectangular object results in circular shapes only; while packing the circular object second will generate a far wider range of shapes (see Fig. 4.64), including circles.

3. **Change position of one of the objects**.

   The last design change considers changes in position and transforms the initial position of an object into a final position that complies with the overlap rules (using the position changes laid out in Section 4.4.5), whilst leaving shape unchanged.

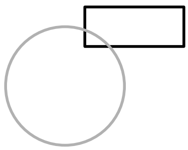


Fig. 4.65 – Prohibited overlap between circle and rectangle

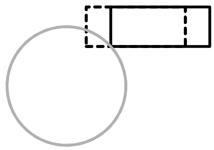


Fig. 4.66 – Removed overlap by changing position of rectangle

To study the effect of sequence, assume situation 1. The initial position of an object is determined by input parameters, e.g., as in Table 4.6 and that this initial position results in unwanted overlap. To remove, it, a change in position is used, i.e., changes the values of input parameters determining the object's position. Consider situation 2, however. In this case the initial position of the object is adjusted such that it co-indices with the final position achieved in situation 1, using overlap management. In this second situation, no overlap management is necessary, and the same position results as in situation 1.

The ability to generate the same position with and without overlap management means that the effect of sequence can be removed.

A simple example shown in Figs. 4.65 and 4.66 helps to illustrate this. They show the positions of the rectangular object before and after overlap management. If the range of initial positions is such that the position after overlap management in Fig. 4.66 can also be created as an initial position by attributing appropriate values to input parameters, the effects of sequence become irrelevant.

Two conclusions can be drawn on the impact of sequence from the application of the three types of design changes.

- **Sequence of application does not restrict the range of configurations** when overlap management changes parameter values, e.g., those governing position and or shape. In those cases, it is possible to remove the effects of sequence provided the input parameter values can attain the same values with and without overlap management. Both the position changes and shape changes using an object's own shape function can therefore be applied without considering the sequence of application.

- **Sequence of application will restrict the range of configurations** when overlap management changes the number, type and meaning of parameters describing an object's shape. This occurs when conforming an object's shape to the shape of another objects. In such cases, one sequence may result in a range of shapes that another sequence cannot generate.

This means that one has considerable freedom when considering the sequence of packing for the envelope object, hard objects and free space objects (see Table 4.2), while sequences matters considerably for subdivision objects, soft objects and connection objects (again, see Table 4.2). Still, even though sequence does not restrict the range of configuration for the envelope, hard objects and free space objects, it might still affect the yield of the search process, which is discussed below.

4.5.4 Does sequence affect the yield of the packing process?

Section 3.2 discussed how the search algorithm alters parameter values in order to find suitable ship configurations. Section 3.4.2 subsequently explained how achieving acceptable yield may be problematic, i.e., the search process generates an insufficient number of useful designs to investigate the many options and trade-offs under consideration. The sequence of packing might influence yield and it is therefore investigated to find appropriate sequences.

To this end, a simplified version of the packing approach presented in this Chapter was developed. It packs a single envelope object and twenty hard objects (all of rectangular shape). All hard objects were prohibited from overlapping with each other, but were required to overlap with the envelope. The envelope object, defined by a simple parabola, uses the shape change using an object's own shape function (described in Section 4.4.5) to manage overlap, while the hard object used changes in positions to comply with the overlap rules (also described in Section 4.4.5). The envelope's initial shape as well as the hard objects' initial x and z-positions were varied by input parameters using the search algorithm selected in Chapter 5.

Special feature of the simplified packing approach is the ability to use an input parameter to change the instance at which the envelope is packed. This makes it possible to cover the range of interactions between envelope and hard objects from pure 'packing': which sees the hard objects change position within an envelope that does not changes shape or size during the packing process (as in *Nick [88]*), to pure 'wrapping': in which all hard objects are packed first before wrapping the envelope object around them (as in *Andrews [3]*).

Fig. 4.67 shows an example where the envelope is packed as the third object. The upper image shows two hard object packed first. The middle image shows the initial envelope shape and the final envelope shape that results from overlap management, i.e., it is wrapped around the two hard objects. Subsequently, hard objects no. 3 to 20 are packed, resulting in the lower image in Fig. 4.67.

The search algorithm's ability to vary between packing and wrapping means it can figure out during the search process how sequence affects the range of configurations as well as the yield. Its findings were subsequently confirmed by changing the instance the envelope was packed manually. The search process was then repeated to cover all twenty-one instances at which the envelope can be packed.

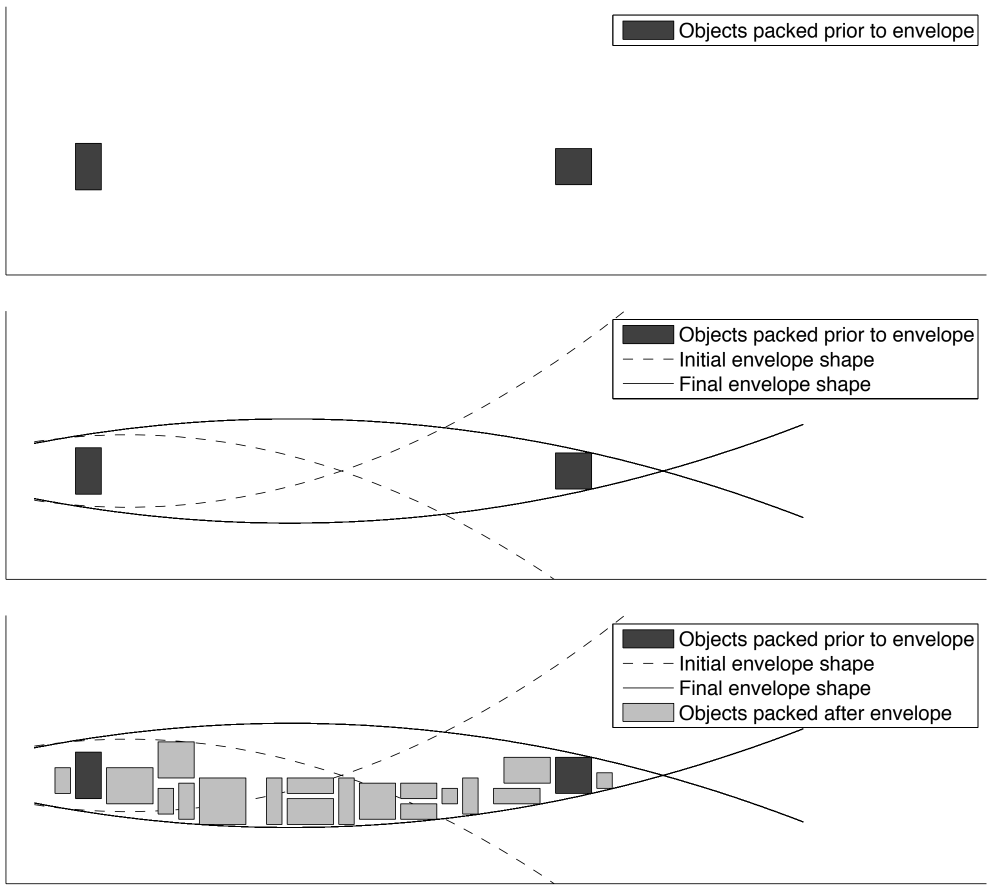


Fig. 4.67 – Process used to combine packing and wrapping: first, place two hard objects, second, wrap the envelope around the two objects, and third, pack all remaining hard objects

The sequence of hard objects was also studied. They were either packed from large first to small last, or viceversa. The order of packing of hard objects was changed manually and left unchanged during the search process.

The search algorithm used two objectives to generate a so-called Pareto-front (*Pareto [93]* and Section 5.2.3): minimise enclosed area of the envelope and maximise the instance the envelope was packed (later is better). This Pareto-front illustrates when the enclosed area of the envelope needs to increase in order to pack the envelope at a later instance. Simply put, the search process tries to 'wrap' the envelope as much as possible (which will give large enclosed areas, as the envelope does not constrain the hard objects' positions to a coherent 'ship-like' shape with appropriate dimensions), whilst still trying to reduce enclosed area.

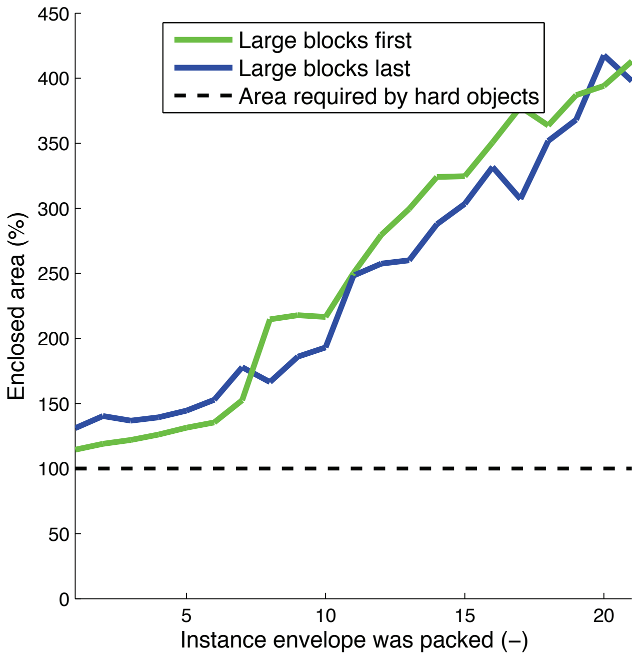


Fig. 4.68 – Impact of sequence on packing density: large first and large last

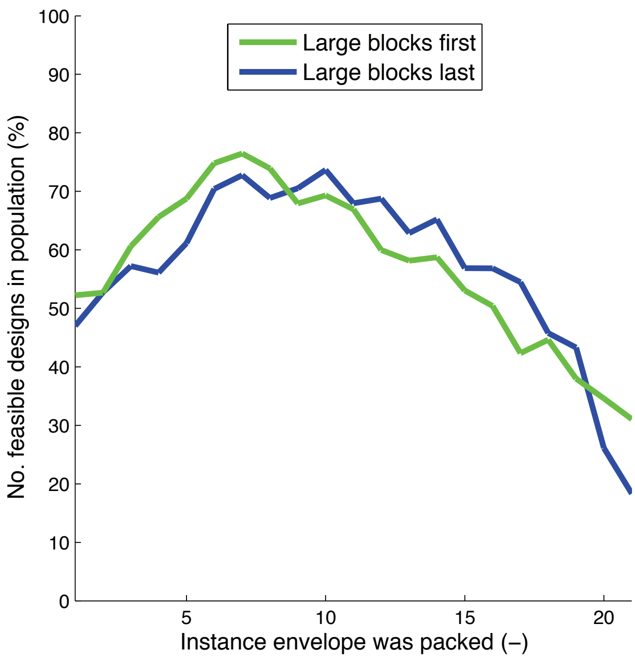


Fig. 4.69 – Impact of sequence on number of feasible designs: large first and large last

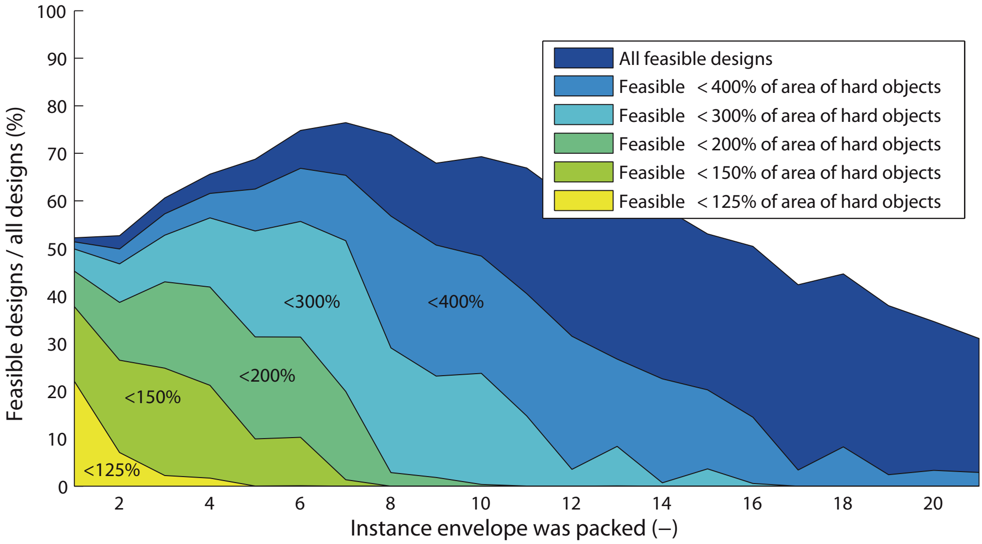


Fig. 4.70 – Impact of sequence on number of feasible designs: large first

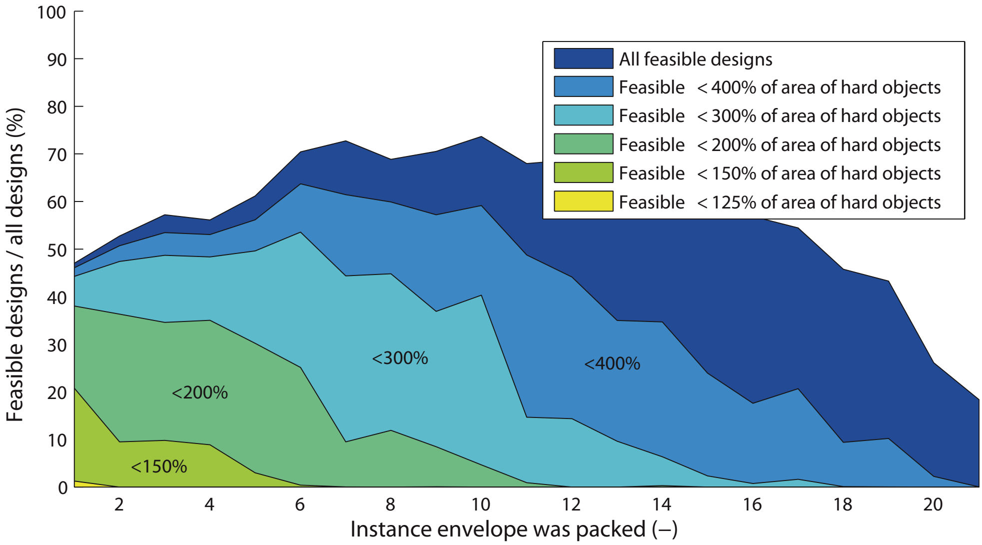


Fig. 4.71 – Impact of sequence on number of feasible designs: large last

The test case shown in Fig. 4.67 to 4.71 is intended primarily to illustrate whether and how the theoretical indifference of sequence for the interaction between envelope and hard objects identified in Section 4.5.3 would be affected by the application of a search algorithm, with a particular focus on the number of useful designs, i.e., the yield, of the search process. Soft objects, free space objects, subdivision objects and connection objects were not used in the test case. For soft objects, subdivision objects and connection objects, Section 4.5.3 already established situations in which the effects of changes sequence could not be removed. Furthermore, it was assumed that the range of sizes and the order of packing would make hard objects representative for other object types. Lastly, the ability to conform shape used by soft objects and connection objects simplifies the packing problem considerably, as these shape changes enable such objects to occupy positions unsuitable for hard objects.

The results from numerous search efforts with the simplified packing approach are shown in Figs. 4.68 to 4.71. They use the results from the search effort where the instance at which the envelope was packed was altered manually. The search runs were repeated ten times to improve reliability. The minimum value obtained over all ten runs was used by Figs. 4.68 to 4.71.

Fig. 4.68 shows the Pareto-front with the smallest enclosed areas (as a percentage of the area required by the hard objects) plotted against the instance the envelope was packed. Figs. 4.69 to 4.71 show the percentage of designs that comply with the constraints (as a percentage of the total number of designs) plotted against the instance the envelope was packed.

The following conclusions are drawn with regard to a suitable sequence of packing.

- Fig. 4.68 illustrates that packing the envelope as the very first object, i.e., ‘packing’ in a classic sense (as in *Nick [88]*), yields the smallest enclosed area and, as a result, the largest range of enclosed areas. Fig. 4.68 only shows minimum area; larger envelope sizes were also found. Packing the envelope as the last object, i.e., ‘wrapping’ according to *Andrews [3]*, yields the largest enclosed areas and a somewhat smaller range of enclosed areas.

  The difference between the sequence of packing of hard objects (large first versus large last) is much smaller. Both packing large first and large last can give a lower enclosed area, depending on when the envelope is packed (the blue and green lines cross twice in Fig. 4.68 and 4.69; no investigation was made into what is causing this). The smallest enclosed area is found by packing the envelope first, followed by the hard objects ordered from large to small.

- Fig. 4.69 shows a more complex trend. When packing the envelope as the very first object, the number of designs that complies with all constraints lies around 50%. It increases subsequently to around 70% when packing the envelope slightly later, before reducing to 20% when the envelope is packed as the last object. This appears to indicate that the yield of the packing process can be increased by using a mix between packing and wrapping, i.e., by packing the envelope early on, but not as the very first object. Note, this trend does not depend on the sequence of packing of the hard objects.

- Fig. 4.70 and 4.71 reveal that the trend in Fig. 4.69 is, in fact, even more complex. Both Figs. 4.70 and 4.71 show the number of feasible designs broken down on their enclosed area.

> This reveals the large impact of the sequence of packing the hard objects. Fig. 4.70 shows that packing the envelope first, and subsequently the hard objects ordered from large to small, not only results in the smallest enclosed area, but also that the yield, i.e., the number of configurations with such high packing densities, is considerable (e.g., enclosed area smaller than 150 % of that required by the hard objects). Fig. 4.71 on the other hand shows that packing the hard objects from small to large does yield a negligible number of useful designs.

In summary, the simplified test-case illustrated both 'packing' (*Van der Nat [120]* and *Nick [88]*) and 'wrapping' (*Andrews [3]*) can yield feasible designs and are therefore considered to be valid ways to handle the interaction between the internal arrangement and envelope shape and size. Nonetheless, the simplified test-case shown in Fig. 4.70 illustrates that packing the envelope first before packing other objects in the order from large to small provides both the largest range of enclosed areas and the largest yield of useful designs (loosely defined by having an enclosed area smaller than 150%). It is this sequence that will form the basis for the sequence of packing proposed in Section 4.5.8.

Note, these conclusions are based on the results in Fig. 4.68 to 4.71 and as such, use a data-set that is limited in scope (for instance, no variation of the hard objects' sizes was carried out). Nonetheless, the example is deemed to have enough similarity with the three-dimensional packing-based ship description developed in this Chapter that the conclusions are considered sufficiently valid to help establish the sequence of packing for the applications in Chapter 6.

4.5.5 Using sequence to remove shape changes using its own shape function

Section 4.5.3 illustrated how two of the three design changes are unaffected by sequence (under specific conditions). This means one has some freedom to choose whether to actually use these design changes or, instead, to forgo their application in order to simplify the implementation of the packing process.

Section 4.4.5 presented the three design changes for overlap management: position changes (see Fig. 4.39), shape conformation (which means one object 'borrows' part of the shape of another object; see Fig. 4.37) and shape changes using an object's own shape function (see Fig. 4.35).

Of the three types of design changes, the position change is not influenced by sequence but used by five object types (see Table 4.2) and therefore its implementation cannot be avoided. Similarly, shape conformation is used often, and also strongly influenced by sequence; both are good reasons to implement this design change as well.

The shape change using an object's own shape function differs from the two other design changes. First, the shape change using an object's own shape function is only used by one object type: the envelope (see Table 4.2). Second and most importantly, Section 4.5.3 argued that the range of shapes that can result from this design change can be made independent of the sequence in which this design change is applied, provided that two conditions are met.

The first condition is that the envelope is packed as the very first object, the second condition is that the range of initial shapes of the envelope (i.e., prior to overlap management) is similar to the range of final envelope shapes that would result from overlap management using the object's own shape function.

Design changes	Change position	Change through shape function	Conform own shape	Change envelope shape
Envelope	not relevant	**no**	yes	not relevant
Subdivision	yes	no	yes	no
Hard	yes	no	no	no
Soft	yes	no	yes	no
Free space	yes	no	no	yes
Connection	not relevant	no	yes	no
Logical object	yes	no	yes	yes

Table 4.10 – Design changes per object type after considering packing sequence (updated version of Table 4.2)

Crucially, both conditions can be met; the former by deciding to pack the envelope first, the latter by allowing a sufficiently wide range of parameter values to vary the envelope's initial shape. As a result of this, one could therefore choose to forego the use of this particular design change.

Not using this particular design change is beneficial. It will save considerable effort, because developing and implementing a useful and practical approach to change shape using an object's own shape function is difficult for real-life three-dimensional hull forms. An advanced -yet practical- implementation is presented by *Harries [56]*. It can adjust the hull shape to enclose user-defined 'hard points' whilst taking hull characteristics such as fairness, required displacement and centre of buoyancy into account (see *Abt et al. [1]* for examples).

It would be possible to integrate the approach proposed by *Harries [56]* in the packing approach presented in this Chapter. However, it is even simpler not to use the shape change using an object's own shape function at all. This can be exploited to remove the need to integrate *Harries [56]*'s approach. Most importantly, it does not reduce the range of ship configurations the packing approach can generate (following the discussion above and in Section 4.5.3).

Therefore, it was decided that the envelope would be packed as the very first object. Simultaneously, it was decided that the envelope will be provided with sufficient changeability of its initial shape, to remove the impact of this particular sequence. Together, the two decisions mean the envelope's initial shape will always become its final shape, as overlap management is not necessary (shape conformation of the topside part does occur later on in the packing process, following Section 4.4.1, e.g., Fig. 4.45). All other objects are packed after the envelope object.

This results in an adjustment of the design changes used by the seven object types from Section 4.4.5 (listed in Table 4.2). The updated version is shown in Table 4.10.

Note, packing the envelope first does not mean that envelope is static during the entire search process. The envelope's initial shape and size will still be varied by the input parameters from the search algorithm (see Table 4.3). As a result of this, the packing-based parametric ship description will be able to vary the ship's main dimensions, e.g., to determine the size required

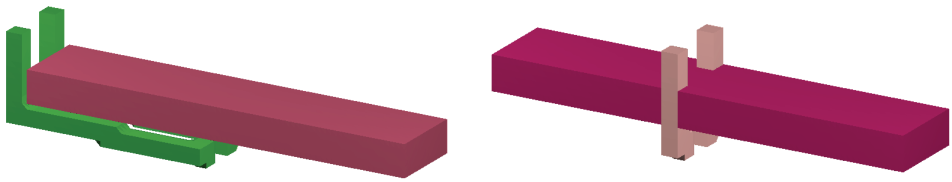


Fig. 4.72 – Connection object conformed to a soft object (left) and viceversa (right)

to accommodate all systems in a feasible ship design.

Chapter 6 will investigate whether this decision to pack the envelope as the very first object restricts the range of ship sizes considered during the search process (see Section 6.4.2).

4.5.6 Other issues affecting sequence: other object types

The parametric ship description also uses object types other than the envelope object and hard object. This means they must also be integrated in a packing sequence. The order of packing is determined by human-defined preferences stating which particular objects are allowed to change which other objects.

For instance, assume one has a large cargo-deck on an auxiliary oiler replenishment and uses an area-based soft object to model it. One also has two uptakes of the propulsion system each defined by a connection object (shown in Fig. 4.72). Table 4.2 states both object types may use shape conformation to manage overlap and, therefore, one has to decide which object is to change shape.

Packing the soft object prior to the connection objects will result in configurations where the uptakes are routed around the cargo deck, leaving a large unobstructed cargo deck (left image in Fig. 4.72). Packing the connection objects first and the soft objects later will result in simpler uptakes, with the soft object conformed around them. This means the cargo deck does have some small obstructions, which could be problematic, while exhaust losses are lower (right image in Fig. 4.72).

A second example would be the packing of hard objects and bulkheads, where different sequences may result in different positions for both the bulkheads and the hard objects. One possible sequence is packing hard objects prior to the bulkheads, i.e., bulkheads positions are driven by the hard objects. Another sequence is placing bulkheads first and hard objects later, which will see the latter's position driven by the bulkhead positions.

Which sequence for a particular pair of objects is preferable depends on the naval architect's assessment, together with the type of service vessel under design. This simple issue: determine which of a pair of objects should change shape or position, is repeated for all objects and must be considered thoroughly when developing the parametric ship description and the sequence of packing.

4.5.7 Other issues affecting sequence: focussing

Focussing, to be discussed in Section 4.6, constrains and controls the positions of objects and therefore interacts with the sequence of packing of objects. The reason is that placing constrained objects late in the packing process increases the chance that other objects (with fewer constraints) will occupy the positions required by the more constrained objects, which would cause the packing process to fail. Adjusting the sequence of packing such that constrained objects are placed earlier should therefore increase the yield of the packing process.

The various ways of focussing influence sequence in different ways. Sometimes, this simply concerns freezing part of the packing sequence. For instance, assume one uses a logical object to link a hard object (a large crane) to a subdivision object (a bulkhead to ensure structural support for the crane). This means that the subdivision object must be packed earlier than the hard and logical objects, in order for the focussing to work. Other ways of focussing, such as restricting the position of an object, e.g., to constrain the bridge between 25% and 75% of the ship's length, are more flexible with regard to sequence. However, even for these types of focussing it will be beneficial to pack the objects that use them early on in the packing process, e.g., to help to increase the yield of the search process.

4.5.8 Proposed sequence

This Section introduces the basic sequence proposed for the packing approach presented in this Chapter. It builds upon the discussion in Sections 4.5.1 to 4.5.7 and forms a reasonable compromise between the different impacts a sequence might have. Furthermore, it assumes one has derived a complete set of objects using the types from Section 4.4.1 that, together, describe the service vessel of interest at the appropriate level of detail.

First and foremost, objects are packed concurrently in small groups which have been defined as 'systems of objects' (following Section 4.5.1). Second, these groups are subsequently packed in a sequential manner, using the following sequence.

- The envelope is packed as the very first system of objects, as discussed previously.
- The decks are positioned next (they are defined as a system of subdivision objects) on which other systems of objects will be placed.
- A few systems with hard and soft objects are placed next. These systems may contain objects that are large, constrained or otherwise difficult to place and thus might require bulkhead positions to be altered (see below). This step can be skipped if no such objects are present.
- Bulkheads are placed (defined by subdivision objects) such that they do not overlap with previous hard and soft objects, but still comply with non-negotiable requirements on reserve of buoyancy (Fig. 4.48).

- All other objects are placed in the order from large and / or constrained, to small and / or less constrained. Furthermore, free space objects are placed fairly early on, to restrict the shape the superstructure part of the envelope may take.

- After all objects are packed successfully, the final shape of the topside part of the envelope is determined by removing those parts not used to enclose or support other objects (resulting a shape such as in Fig. 4.45).

Note that the proposed sequence offers guidance only. Properly establishing a sequence requires several search runs to study the effect of different sequences on yield and the resulting configurations.

As an example, an actual packing sequence is outlined in Chapter 6 and shown in Figs. 6.3 and 6.4.

4.6 Focussing

Focusing enables the naval architect to prevent generating configurations with unwanted positions for systems of objects. Several complementary methods are used to control positions and shapes. They are discussed below.

- Constrain positions and shapes using sequence. This was discussed already in Section 4.5.6 (Fig. 4.72, for instance). Sequence can be used to prevent unwanted shape and position alterations to objects.

- Constrain positions using a system of objects. If one wants to constraint relative positions of multiple objects, one can model them as a system of objects (discussed in Section 4.5.1). This enables one to model a weapon system on a frigate, such as in Fig. 4.25, by combining a free space object (green in Fig. 4.25) with a hard object (white in Fig. 4.25) with a fixed relative position. Together, the two object types correctly mimic the original weapon system they represent.

- Constrain positions using permissible x, y, z-coordinates. The position changes discussed in Section 4.4.5 alter an object's x, y, z-coordinates. One can remove unwanted position changes by constraining the x,y or z-coordinates an object may take. For example, one might restrict a hard object to a deck by prohibiting the object from using z-coordinates that do not coincide with the deck positions. Similarly, one can place an object at the centre-line by restricting the permissible y-coordinates to the specific coordinate required to achieve this position.

- Constrain positions using fixed or bounded initial positions. Section 4.4.5 discussed how overlap management changes an object's initial position to a final position that complies with the overlap rules from Table 4.1. One can use input parameters to change such initial positions (as in Tables 4.6, 4.7 and 4.8), but can also use fixed or severely bounded values. Such fixed positions help focus the configuration.

  For example, assume one wants to place an object representing a crane on the starboard-side edge of the envelope (shown in Fig. 4.73). To achieve this, one can use an initial

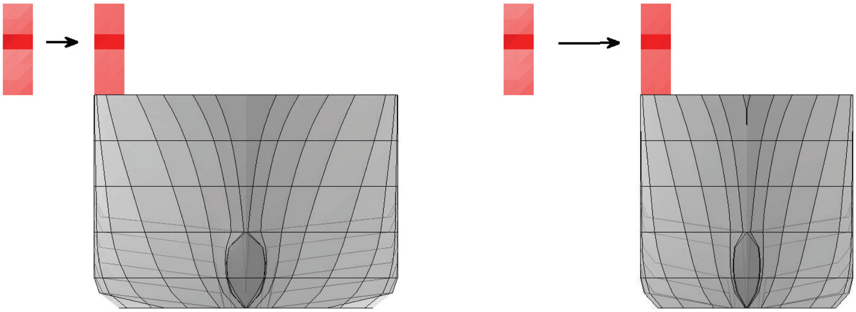


Fig. 4.73 – Same initial y-position for a crane, different hull widths (initial position is outside the envelope, final position on the envelope)

transverse position that lies on the starboard side edge of the positioning space, i.e., outside the envelope object (the left crane in both images of Fig. 4.73). The position change discussed in Section 4.4.5 (Fig. 4.50) will achieve the required overlap by shifting the crane in transverse direction until the required position at the edge of the envelope is reached. The position change then stops, which prevents the crane from being located further inwards than necessary (the right crane in both images in Fig. 4.73). Note, this stopping criterion for position changes was shown in Fig. 4.50. Most importantly, the crane position at the edge of the envelope is achieved regardless of hull width.

- Constrain relative positions using logical objects. The most elaborate and flexible method to constrain relative positions of objects are logical objects. Though briefly discussed in Sections 4.4.1 and 4.4.3, a more elaborate explanation of their workings is warranted. Logical objects use the shape description and design changes of the six physical objects discussed in Section 4.4.1, i.e., one has logical objects that share the features of hard objects, and logical objects that share features with soft objects. Both features are used for overlap detection and management with other objects.

  Where they differ is in the overlap rules. Section 4.4.3 explained how logical objects have bespoke, user-defined overlap rules, which are defined in two steps. First, one defines which previous objects the logical object considers during overlap management. These objects may be one or more of the six physical object types, or other logical objects (such as the blockage around a weapon system in Fig. 4.27). Second, one defines whether the logical object should or should not overlap with those previous objects.

  A simple example in Figs. 4.74 to 4.76 illustrates the workings of logical objects in more detail, using a simplified naval replenishment vessel as an example. Fig. 4.74 shows three soft objects representing fuel tanks.

  Fig. 4.75 shows the same fuel tanks and the current system of objects: a hard object representing a fuel pump (in red) and a logical object (in green). Assume one wants to reduce the length of fuel piping by constraining the relative position of the fuel pump to a location adjacent to one of the fuel tanks. To achieve this, the logical object refers to

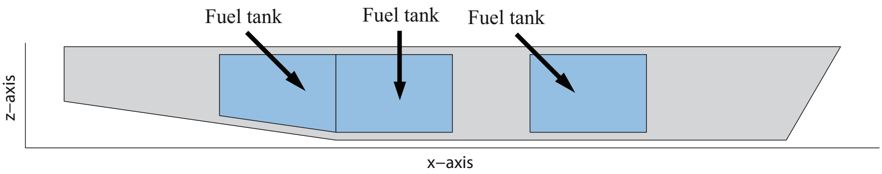


Fig. 4.74 – Three fuel tanks (soft objects shown in blue)

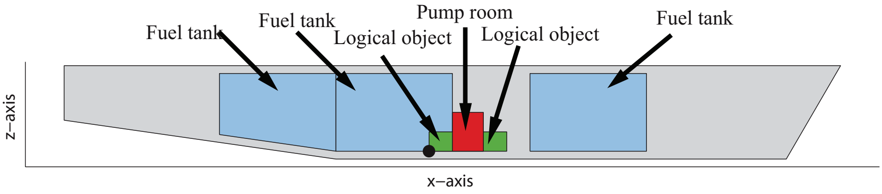


Fig. 4.75 – A fuel pump (in red) placed adjacent to one of the fuel tanks using a logical object (in green)

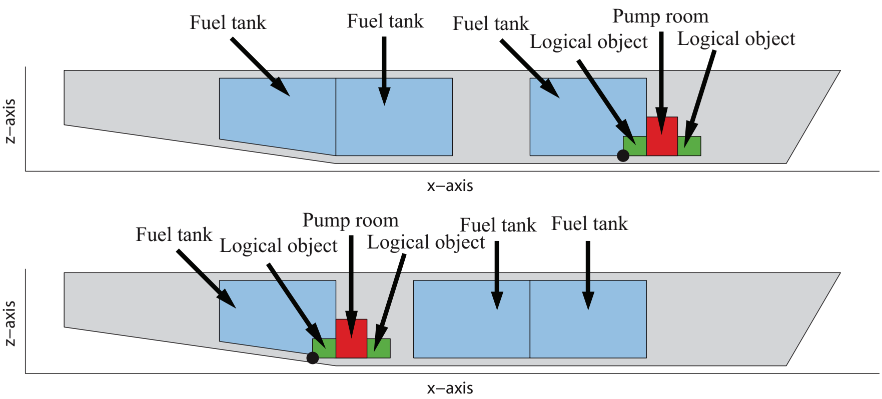


Fig. 4.76 – Different fuel pump (red) positions resulting from different initial positions; the logical object (green) ensures adjacency

the three fuel tanks and is required to overlap at least partially with at least one of them. The red object is prohibited to overlap with the fuel tanks following the overlap rules in Table 4.1. Fig. 4.75 shows one position where the system of objects has the required adjacency, i.e., the logical object (green) has partial overlap with one of the fuel tanks, while the fuel pump (red) does not overlap with the fuel tanks.

Changing the initial position of the system with the hard object and logical object results in a different position for the fuel pump, shown in the upper image in Fig. 4.76, while still achieving the required adjacency (with a different fuel tank, though). Even changes in fuel tank configurations can be handled (lower image in Fig. 4.76).

One can easily formulate other uses for logical objects. For example, one can use them to prevent a combat information centre from being located on top of an ammunition hold,

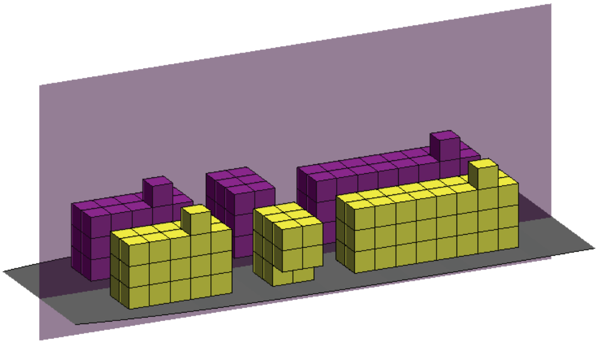


Fig. 4.77 – Original objects and their symmetrical equivalents

or to place it close to the bridge to improve ease of communication during anti-piracy operations.

In summary, though simple in concept, the use of logical objects enables the naval architect to build complex networks controlling the relative positions of objects, while still retaining the flexibility to create different configurations through the use of changes in initial positions. As a result, combining logical objects in a system with other object types enables the naval architect to steer the packing process towards configurations with desirable relative positions of the systems of objects, such as separation, proximity or even symmetry (Fig. 4.77).

- Constrain positions using a positioning space with pre-defined constraints. The last way to constrain objects is by using positioning spaces in that reflect common constraints found on service vessels. In such positioning spaces (discussed in Section 4.4.2), part of the available space is blocked to prevent objects from being placed at these unwanted positions.

  This helps, for instance, to force free space objects to a location above the damage control deck to prevent them from interfering with the ship's watertight integrity. The approach was also used by *Van der Nat [120]* to restrict objects, e.g., to pack fuel tanks in-between the outer hull and pressure hull of a double-hull submarine.

  Figs. 4.78 and 4.79 show some examples (shown as a transverse cross-section of the positioning space and envelope). Blue voxels indicate space that is available to place an object on. The two left-most images show the positioning spaces one can use to pack objects respectively inside and outside the envelope. The left-most image also shows how some voxels (blue) are only partially available; they are used by soft objects with an enclosed volume requirement to improve accuracy without increasing the number of voxels. The two right-most images show, respectively, an unconstrained positioning space and one that only has space available outside the hull part of the envelope (right-most image). This latter positioning space prevents the free space object from adjusting the envelope shape such that it affects the watertight integrity. Positioning spaces similar to those in Figs. 4.78 and 4.79 were used to constrain objects to a location inside the ship's hold or within a double hull (for fuel tanks).

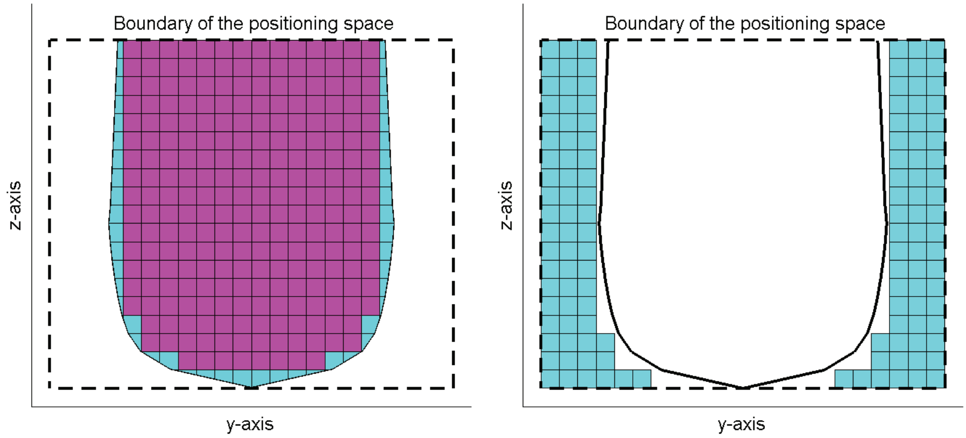


Fig. 4.78 – Positioning spaces with pre-defined constraints to focus object positions

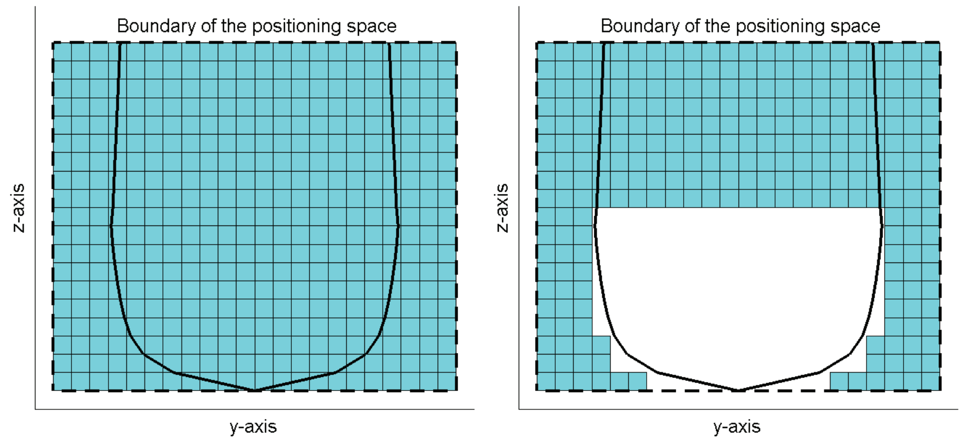


Fig. 4.79 – Positioning spaces with pre-defined constraints to focus object positions

Note that the methods to focus the packing process are complementary to the packing requisites discussed in Section 4.4, i.e., one can use them to focus the packing process, but only if necessary. This means the methods discussed above do not simplify the implementation of the packing process (to be avoided following Section 3.5.2). Instead, they offer a flexible method of control added onto the overall packing approach to steer the packing process towards particular configurations; a method that can be used and adjusted rapidly when required (see the discussion on yield in Section 3.4.2).

4.7 Packing process

The previous Sections introduced the requisites of the packing approach developed in this Chapter. They covered the object types and the positioning space, the overlap rules and the design changes used to enforce them, and argued the need to pack systems of objects with fixed relative positions in a sequential manner.

What remains is an overview of the packing process that explains how all requisites together integrate into one coherent packing process that enables input parameter values to change the entire ship descriptions, which is discussed in this Section.

4.7.1 Outline of the packing process

The packing process is shown in Fig. 4.80 and consists of two steps: definition and packing. Both are introduced below. Note, it assumes a ship description has been invented and assembled following steps one and two from Fig. 3.6. Fig. 4.80 only shows the steps performed when updating the parametric ship description during the search process, i.e., step three in Fig. 3.6.

```
Definition:

        Determine initial shapes and positions of all objects
        in all systems, using pre-defined, static input from
        the naval architect, input parameter values from search
        algorithm and preliminary performance predictions based
        on assumptions (see Section 5.5).

        Establish the packing sequence and the x,y and z-coordinates
        the systems may use.

Packing:

        For System = 1:Number of systems

             a. Retrieve information for current system of objects.
             b. Build up-to-date positioning space for each object
                in current system using the voxels of prior systems
                of objects combined with the overlap rules.
             c. Apply overlap management to pack current system of
                objects whilst complying with relevant overlap rules
                and user-defined constraints for all objects within
                the current system.
             d. After packing current system

            1. If system can be packed: store results.
            2. If system cannot be packed: fail elegantly and
               stop packing process.
        Next
```

Fig. 4.80 – Outline of the packing process

Definition

The first step shown in Fig. 4.80: definition, uses static input (human inventions) together with input parameter values to determine the initial shape and initial position of all systems of objects (notice the resemblance with the 'sizing' step used by *Van der Nat [120]*, see Fig. 4.10). Furthermore, additional input parameters may be used to enable parametric changes of other aspects of the configuration, such as the presence of objects, their number and their connectivity (see Section 3.5).

Key issue is that the initial definition of all systems of objects alters when different values of input parameters are used to produce different configurations after overlap management (as discussed in Section 3.5.1).

Also included in the definition phases are the range of x, y and z-positions an object may take, as well as any positioning space with pre-defined constraints they may use (Section 4.6). Once described, the initial definition of the ship, i.e., the initial positions and shapes of all systems of objects are adjusted in the second step: packing.

Packing

The second step shown in Fig. 4.80: packing, uses the design changes from Section 4.4.5 to enforce the overlap rules proposed in Section 4.4.3. It does so in four steps, repeated in a sequential manner until either all systems of objects are packed successfully or the packing process halts because a system cannot be placed:

1. Retrieve the initial shape and initial position of each object in the current system, as well as the overlap rules that apply to the objects being packed (Table 4.1).

2. Build up-to-date positioning spaces, one for each object in the current system. These positioning spaces contain all prior objects the packing process must consider when using overlap management to help find a suitable position and shape for each object in the system that is currently being packed.

3. Manage overlap for the current system of objects being packed (using the design changes for overlap management discussed in Section 4.4.5) to search for object shapes and positions that comply with the overlap rules. If multiple positions are suitable, the one located closest to the system's initial position is used.

4. Store the positions and shapes of all objects in the system, if overlap management is successful, or stop the packing process automatically, if overlap management fails.

The definition step in Fig. 4.80 (discussed above) speaks for itself. However, two of the four parts of the packing step warrant further explanation. They are building updated positioning spaces and what happens after attempting to packing the current system. Both are discussed below, in Section 4.7.2 and 4.7.3, respectively.

4.7.2 Building updated positioning spaces

The design changes discussed in Section 4.4.5 are used to manage the overlap between the objects in the current system and all objects packed prior to the current system. To enable the application of overlap management, the packing process must be aware of any voxels occupied by these prior objects.

This means the role of the positioning space must be considered in more detail. Section 4.4.2 explained the positioning space's purpose is to hold all objects. It starts out as an empty box that gradually fills up by applying the sequential packing process outlined in Section 4.5. It is this gradual filling that allows the positioning space to be updated with all prior objects relevant for the overlap management of the current system of objects.

More specifically, it is necessary to create as many updated positioning spaces as there are objects in the system currently being packed, with each positioning space tailored to a particular object in the current system. The reason is that a system of objects can consist of different object types, each with different demands with respect to enclosure and different pre-defined constraints (Section 4.5.1), differences that cannot be accommodated using a single positioning space.

An updated positioning space is created in three steps, which are introduced below. They result in an updated positioning space for a single object and are repeated for every object in the system of objects under consideration.

1. Step 1: the naval architect selects a positioning space with the appropriate pre-defined constraints when developing the parametric ship description. Such positioning spaces were discussed in Section 4.6 (examples are shown in Figs. 4.78 and 4.79). If one does not need to constrain the position of an object this way one simply selects an unconstrained positioning space (as shown in the left image in Fig. 4.79). The packing process retrieves the relevant positioning space chosen by the naval architect for the particular object and uses it for the next steps.

2. Step 2: the packing process reviews all objects packed prior to the current object to check if they are required to or prohibited to overlap (objects with permissible overlap are ignored). The overlap rules follow either from the object type as listed in Table 4.1, or, if the current object is a logical object, from the objects the logical object refers to (see Section 4.6). Next, required overlap is converted into prohibited overlap, using a simple rule: if Object 1 must overlap with Object 2, Object 1 must simultaneously prevent overlap with the space unoccupied by Object 2. This enables one to handle both prohibited and required overlap in the same manner. Lastly and most importantly, those voxels that are unavailable to pack the current object are blocked to prevent to current object from using them.

3. Step 3: Merely blocking the space occupied by prior objects is not sufficient to handle the complex interaction between the envelope and other object types. The reason is illustrated by Section 4.4.5, and in particular Figs. 4.42, 4.43 and 4.44. Shape changes applied to the envelope must take into account the need of objects to be enclosed by, be supported by, or be located outside of, the envelope. As such, Step 2 (blocking space occupied by prior objects) must be complemented by additional measures that prevent the shape changes of the envelope from inadvertently compromising this task.

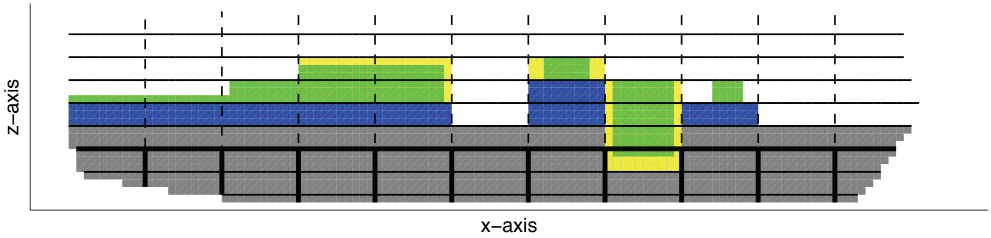


Fig. 4.81 – Objects (in green), together with the internal and supporting compartments (in yellow and blue, respectively)

This is achieved by labelling compartments (enclosed by two decks and two bulkheads) either 'internal', 'supporting' or 'exterior'. The label 'internal' means the compartment contains one or more objects that are required to overlap with the envelope (the hangar in Fig. 4.42). The label 'supporting' means the compartment itself may be empty, but has one or more objects located directly on top of the compartment that required support to prevent them from floating around in the free air[4] (the flight deck and hangar in Fig. 4.44). The label 'exterior' means the compartment contains objects that are to be located outside the envelope, e.g., free space objects, requiring the envelope to be removed in-between the bulkheads and decks. Fig. 4.55 showed an example of a free space object. Note how the envelope shape is adjusted beyond the direct extend of the free space object. The shape change starts and ends at the deck and bulkhead positions closest to the edge of the free space object.

A more elaborate example is shown in Fig. 4.81 dealing with compartments labelled 'internal' or 'supporting'. It shows several objects (in green) together with the compartments labelled 'internal' and 'supporting' that enclose them (in yellow) or support them (in blue), respectively.

With the updated positioning space available, the packing process can pack the current system of objects by placing them at their initial position with their initial shape, and manage overlap using the design changes from Section 4.4.5. This results in a verdict as to whether all objects of the system can be placed successfully. What needs to happen next is discussed in the next Section.

4.7.3 After attempting to pack the current system: storage and handling failure

The sequential packing of systems of objects (Section 4.5) uses the design changes from Section 4.4.5 to alter object positions and shapes to comply with the overlap rules. In many cases, the overlap management will succeed for the current system of objects. In some cases, however, overlap management will be unable to enforce compliance with overlap rules. Both situations must be handled without human interaction and are therefore discussed in more detail below.

[4]This is a prime example of how stupid software actually can be; no naval architect would make such a mistake.

- Overlap management is successful. Overlap management is successful if all objects in the current system can be placed with an acceptable shape at a position that complies with the overlap rules. In that case, one must store the voxels occupied by each object to use them later on, e.g,. to predict performances, or to study the configuration in more detail.

  Storage is especially important. It is necessary to built the updated positioning space for subsequent systems of objects (see Section 4.7.2). The voxels of each object at its final position are therefore stored, together with all information that the packing process required to find this position (e.g., initial position, initial shape, overlap rules).

  After storage, the packing process continues with the next system of objects until either all systems are packed successfully, or the process fails because overlap management is unsuccessful (Fig. 4.80). If all systems of objects are placed successfully, the storage enables the software to re-create the configuration by visualising the voxels of the design, which is considerably faster than re-running the entire packing process.

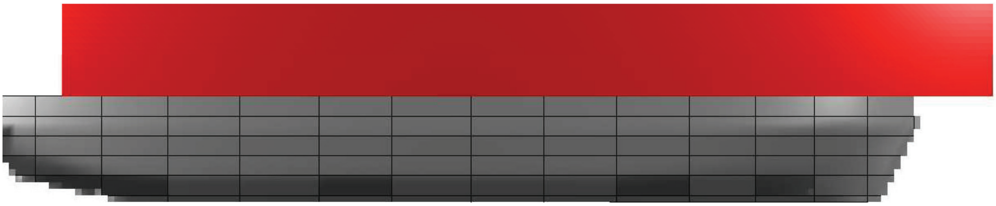


Fig. 4.82 – Free space object preventing other objects from being placed on the main deck

- Overlap management is unsuccessful. In some cases it is not possible to find positions and shapes that comply with the overlap rules. This raises two issues. First, what may cause such failures? Second, how should the packing process handle these failures.

  Govaarts [49] identifies two main reasons why the packing process fails. Either insufficient space is available, i.e., the ship is too small, or the packing of earlier systems of objects has yielded ‘inefficient’ configurations, i.e., configurations that are not necessarily too small, but in which the position of earlier systems of objects is such that they block all positions suitable for the current system of objects.

  Placing a forward firing gun[5] at the stern of a warship offers an example of an ‘inefficient’ configuration. Fig. 4.82 shows how its accompanying free-space object (shown in red) prevents other systems (such as a hangar) from being placed on the main deck, causing the packing of such objects to fail.

  The packing process stops entirely, without human interaction, when a system of objects cannot be packed. The number of the system where packing fails is used to calculate **the number of systems that could not be packed in this particular attempt**. This latter number gives a numerical measure stating how successful the particular initial positions and initial shapes of the objects were, in combination with overlap management, in generating a configuration in which all systems of objects could be placed at a position that complies with the overlap rules. As such, attempts where the number of systems that

[5] The gun is defined by two objects: a hard object, and a free space object that extends to the forward edge of the positioning space

could not be packed is zero or very low are regarded as more successful than those attempts where the number of systems that could not be packed is higher. This means **the number of systems that could not be packed, is suitable for use as a constraint for the search algorithm** (see Chapter 5). It helps to guide the search process towards configurations in which all systems of objects can be packed successfully, thereby increasing the yield.

After stopping with the current attempt, the search process continues with a new attempt, using different values for the input parameters resulting in different initial shapes and positions for the systems of objects.

The reason to stop the entire packing process is straightforward. If a system cannot be packed successfully, one might skip it and continue with subsequent systems (as used in a prototype packing implementation discussed in *Van Oers et al. [129]* and *Van Oers et al. [132]*). This approach becomes fraught with difficulties when using logical objects (discussed in Sections 4.5 and 4.6) as they introduce considerable dependency between systems of objects in the packing sequence. This means skipping a particular system can cause other systems, that rely on logical objects, to refer to a system which was skipped. This means these systems are unable to retrieve the information necessary to pack their logical objects successfully, which would result in configurations that do not conform to the user-defined focussing applied by the naval architect.

Handling successful and failing overlap management without human interaction means the search algorithm can test and alter input parameter values to study their effect on the configuration and the resulting objectives and constraints. Storage enables one to build an updated positioning space for subsequent objects, whilst using the number of systems that could not be packed as a constraint helps to guide the search process towards configurations that can pack all systems of objects such that all objects comply with the overlap rules. Furthermore, storage is crucial for re-creating configurations after the search process.

4.8 A simple application of the packing-based ship description

The results from the previous Sections have been integrated into a packing process in Section 4.7. This process takes a set of user-defined systems of objects, combines them with input parameter values and applies the two design changes for overlap management to convert the resulting initial shape and position of objects into a coherent ship configuration in which all objects comply with overlap rules.

Purpose of the packing process is to achieve large and concurrent changes to the entire ship configuration through changes in input parameter values (Section 4.1). This Section investigates whether the packing process is indeed able to achieve such changes using a parametric description of a simple warship with a limited number of systems of objects.

System No.	Name	No. of objects	Object types
1	envelope	1	envelope (1)
2	decks	-	subdivision (decks)
3	propellers	2	hard (2)
4	propulsion plant	10	hard (6), connection (4)
5	bulkheads	-	subdivision (bulkheads)
6	VLS	2	hard (1), free space (1)
7	fuel tank (PS & SB)	2	soft (1), logical (1)
8	bridge	4	hard (1), free space (1), connection (1), logical (1)
9	flight deck	2	hard (1), free space (1)
10	hangar	3	hard (1), connection (2),
11	radar mast	4	hard (1), free space (1), connection (1), logical (1)
12	uptakes	2	hard (1), connection (1)
13	downtakes	2	hard (1), connection (1)
14	forward gun	3	hard (1), free space (1), logical (1)

Table 4.11 – Overview of the systems of objects used in the baseline configuration and the two variations

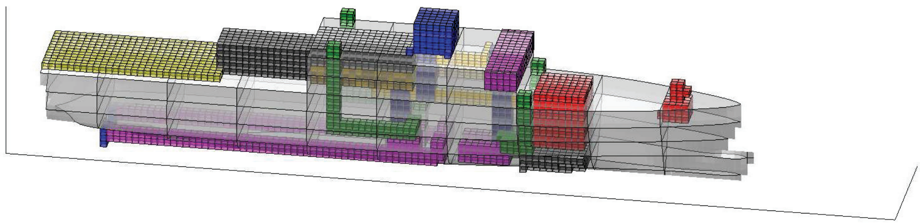


Fig. 4.83 – Baseline configuration of the simplified frigate

Baseline configuration

The baseline configuration is shown in Fig. 4.83 and results from the set of objects from Table 4.11 with the input parameter values from Table 4.12. Note that most initial y-positions (and some x and z-positions) are static and therefore not listed in Table 4.12. The sequential packing process is shown in Fig. 4.84.

Most systems in Table 4.11 and Fig. 4.83 are self-explanatory; some warrant a brief discussion, though.

- System no. 4, the propulsion plant uses six hard objects to model two diesel engines, two gas-turbines and two gearboxes. The two connection objects connect the gearboxes to the propellers of System no. 3. Furthermore, the propulsion plant is packed prior to positioning the bulkheads and as such influences bulkhead spacing.

- The VLS, bridge, flight deck, radar mast, uptakes, downtakes and the gun (Systems no. 6, 8, 9, 11 to 14) all use a hard object in combination with a free space object (the latter to safeguard the interface with the outside world). The logical objects some of these systems employ ensure that they partly overlap with bulkheads (Systems no. 8, 11 and 14), which is useful to provide structural support. A logical object also places the bridge's forward edge onto a bulkhead.

- Several systems have a connection to the network of passageways and staircases. For example, the propulsion plant (no. 4) has two connection objects to this end: one to reach the diesel engines, another to reach the gas turbines. The other systems are the bridge (no. 8), hangar (10) and the radar mast (11). Moreover, the hangar floor itself is not only connected by a connection object, but also doubles as one.

- The hangar is connected to the flight deck by a connection object that mimics a transport route. Similarly, the uptakes and downtakes are connection objects representing large ducts. Moreover, these latter connection objects are combined with hard objects into several systems of objects (one hard object and one connection object per system). The reason is that this will allow overlap management to change both the position of one end of the connection object, as well as the shape of the connection object. The starting point of the connection object is fixed; it is defined by the hard objects representing the propulsion plant (System no. 4).

One of the parameter values in Table 4.12 warrants further attention. Comparing system number 1: the envelope, with system number 6: the vertical launch system, reveals that the VLS' initial position (at $x = 140$ m)[6] is located outside the envelope (which is 110 m long).

This is yet another example of a situation shown earlier in Fig. 4.51 and 4.73, where a system's initial position was also located outside the envelope. Requiring the VLS to overlap with the envelope, and applying the position change discussed in Section 4.4.5 results in the configuration in Fig. 4.83, where the VLS is properly accommodated (partly inside and partly on top of the envelope).

Fig. 4.84 shows how the sequential packing process packs all systems of objects from Table 4.11 into a coherent ship configuration. In this particular case, overlap management is successful, i.e., all objects can be packed while complying with the overlap rules.

Changing envelope and subdivision

Obviously, one needs changes in input parameter values to test whether the description in Fig. 4.83 is actually able to undergo large changes to the entire ship description. These changes are made to two different groups of parameters. First, those parameters governing envelope shape and size and, subdivision positions (deck height and bulkhead spacing) are changed. Table 4.13 lists the changes in parameters. The resulting configuration of the ship is shown in Fig. 4.86; the packing sequence is shown in Fig. 4.85.

[6]The origin at $x = 0, y = 0, z = 0$ lies at the centre-line of the ship, in the horizontal plane coinciding with the keel, directly underneath the aft-most part of the envelope.

System No.	Name	No. parameters	Parameters will change	Value	Unit
1	envelope	4	hull type	1	(-)
			length over all	110	(m)
			maximum beam	16	(m)
			height of the hull	11	(m)
2	decks	2	height of tanktop	1	(m)
			number of DC deck	6	(-)
3	propellers	-	-	-	-
4	propulsion plant	1	initial x-position	60	(m)
5	bulkheads	2	initial spacing	13	(m)
			initial shift	-5	(m)
6	VLS	2	initial x-position	140	(m)
			initial z-position	6	(m)
7	fuel tank (PS SB)	1	initial x-position	100	(m)
8	bridge	2	initial x-position	70	(m)
			initial z-position	7	(m)
9	flight deck	2	initial x-position	1	(m)
			initial z-position	3	(m)
10	hangar	3	initial x-position	38	(m)
			initial z-position	6	(m)
11	radar mast	2	initial x-position	60	(m)
			initial z-position	8	(m)
12	uptakes	2	initial x-position	70	(m)
			initial z-position	7	(m)
13	downtakes	2	initial x-position	40	(m)
			initial z-position	8	(m)
14	forward gun	2	initial x-position	140	(m)
			initial z-position	1	(m)

Table 4.12 – Input parameter values for the baseline configuration

The effect of the parameter changes in Table 4.13 is shown by comparing Fig. 4.83 to 4.86.

Main differences, obviously, include a larger envelope of different shape (the latter due a change in a discrete parameter controlling envelope shape), larger bulkhead spacing made possible by the increase in length, which increases reserve buoyancy considerably. The damage control deck was lowered by one deck-height for the same reason. The changes in envelope and subdivision affect the final positions of systems as well (initial positions were left unchanged, i.e., Table 4.13). For instance, the propulsion plant, the gun and the VLS have moved forward, while the change in bulkhead positions enabled the hangar to move slightly aft. The change in system positions resulted in different shapes of connection objects. For instance, propeller shafts increase in length, and both uptakes and downtakes have a different shape. Finally, superstructure shape also changes to accommodate changes in system positions.

Summarised, the parameter changes in Table 4.13 and the configuration in Figs. 4.86 and 4.85 illustrate that the packing approach can change envelope shape and subdivision positions using input parameter values and, most importantly, can establish their impact on object positions.

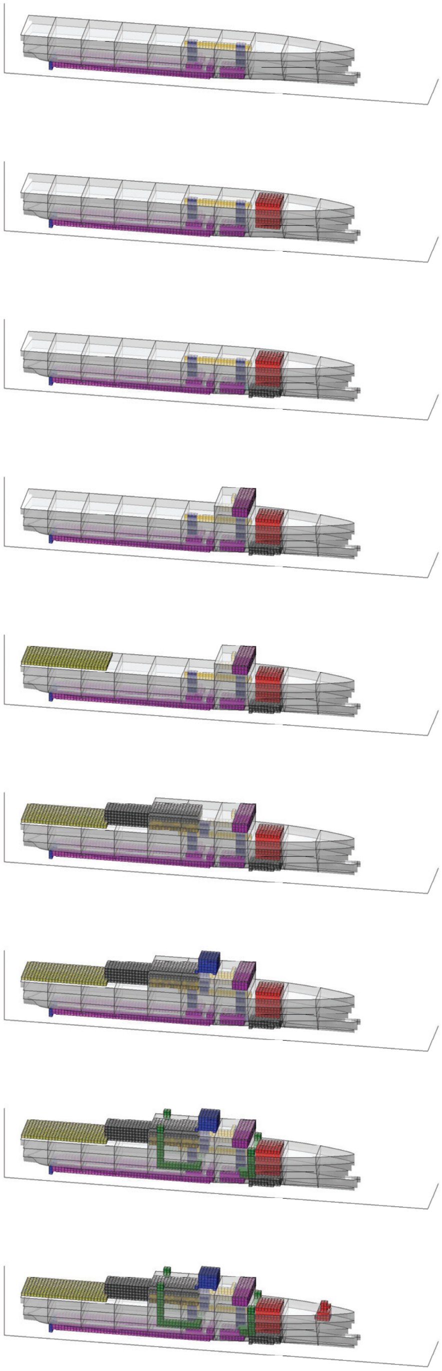


Fig. 4.84 – Baseline configuration (based on Table 4.12)

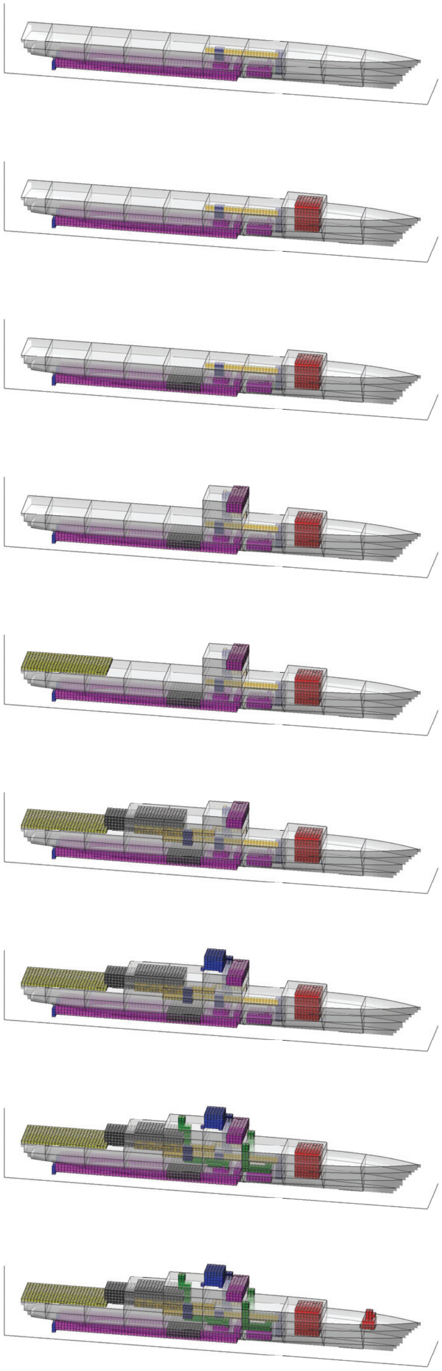


Fig. 4.85 – Configuration with different envelope and subdivision (based on Table 4.13)

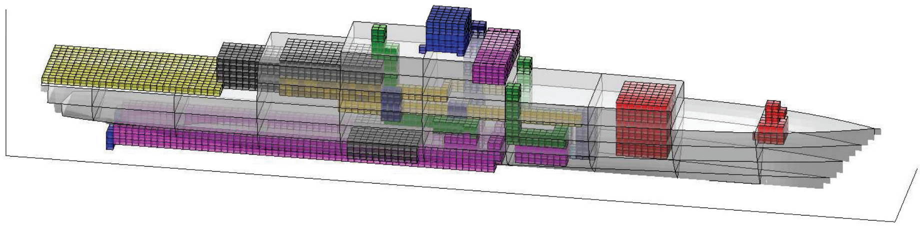


Fig. 4.86 – Configuration of the simplified frigate after altering envelope and subdivision parameters (Table 4.13)

System No.	Name	Parameters will change	Original value	Changed value	Unit
1	envelope	hull type	1	2	(-)
		length over all	110	130	(m)
		maximum beam	16	14	(m)
2	decks	height of tanktop	1	2	(m)
		number of DC deck	6	5	(-)
5	bulkheads	initial shift	-5	0	(m)

Table 4.13 – Table with parameter changes to envelope and subdivision

Changing initial positions

Input parameters can also alter the initial positions of objects. The changes made are shown in Table. 4.14; the resulting configuration is shown in Fig. 4.87. The sequence of packing is shown in Fig. 4.89.

The effect of the parameter changes in Table 4.14 is shown by comparing Fig. 4.83 to 4.87.

Main differences do not concern envelope shape and size but the relative positions of systems. The flight-deck has moved forward, the hangar slightly aft, the bridge and VLS far aft. All these systems have altered initial positions. The radar's initial position has not changed, however, yet overlap management still moves its final position further aft to achieve the required support by the superstructure. Again, superstructure shape changed considerably, to offer support and enclosure to the systems with altered positions.

Summarised, the changes in initial positions do lead to a completely different configuration of systems with an envelope of similar size. Moreover, overlap management is able to find suitable positions even for those systems that did not have their initial position changed, e.g., the radar system.

System No.	Name	Parameters will change	Original value	Changed value	Unit
4	propulsion plant	initial x-position	60	10	(m)
6	VLS	initial x-position	140	1	(m)
		initial z-position	6	7	(m)
8	bridge	initial x-position	70	1	(m)
		initial z-position	7	8	(m)
9	flight deck	initial x-position	1	60	(m)
10	hangar	initial x-position	38	18	(m)
12	uptakes	initial x-position	70	40	(m)
		initial z-position	7	9	(m)
13	downtakes	initial x-position	40	1	(m)
		initial z-position	8	6	(m)

Table 4.14 – Table with parameter changes for positions of systems of objects

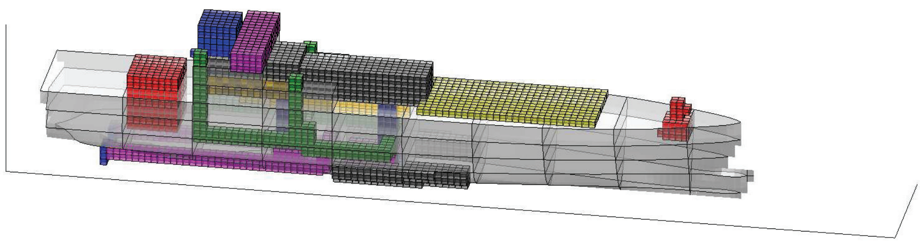


Fig. 4.87 – Configuration of the simplified frigate after changing the objects' initial positions (Table 4.14)

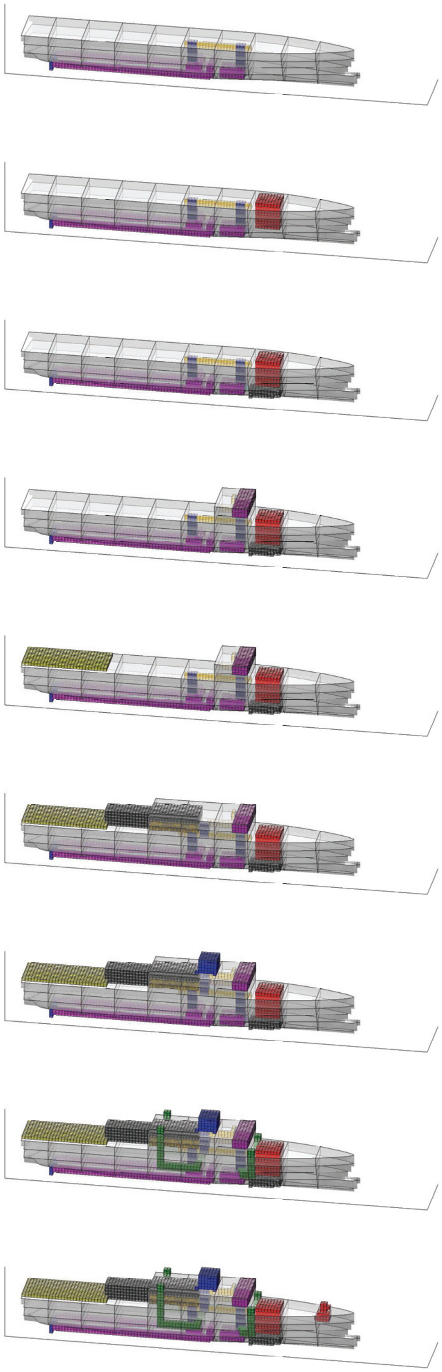


Fig. 4.88 – Baseline configuration (based on Table 4.12)

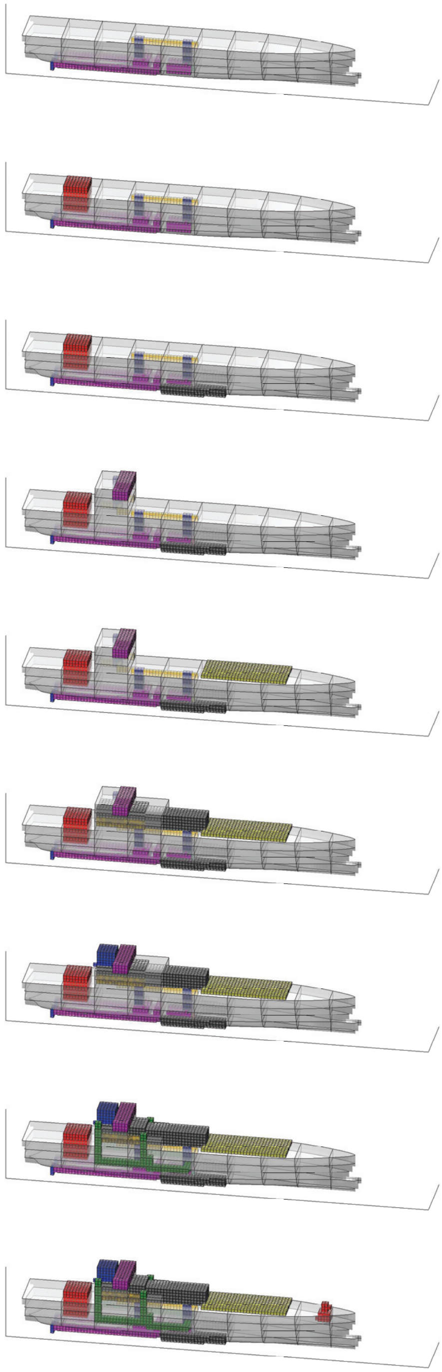


Fig. 4.89 – Configuration with different initial positions (based on Table 4.14)

Summary

The examples discussed previously and shown in Fig. 4.90 illustrate how changes to input parameter values affect the resulting ship configurations.

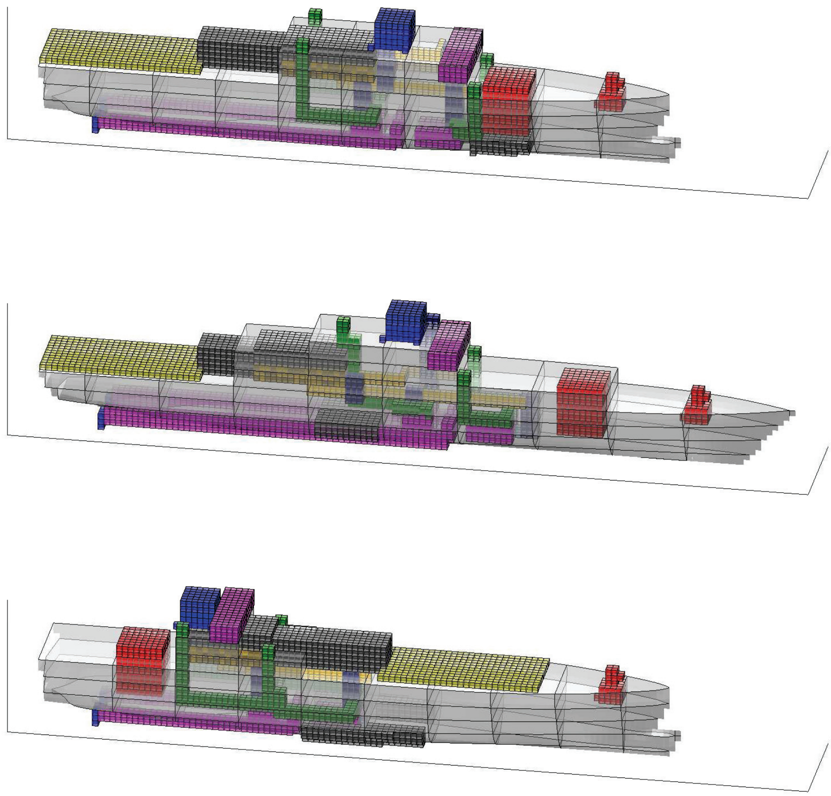


Fig. 4.90 – The original configuration (upper) and two modified configurations that result from changed parameter values (middle, lower)

In combination, these parameter-driven changes are indeed able to alter the entire ship configuration. One might note that changes to envelope and subdivision were treated separately from changes to initial positions, but this was done to better illustrate their consequences on the configuration. Combining them is very simple and Chapter 6 will show configurations that result from the combined variation of envelope shape and size, subdivision, and initial positions of systems.

The discussion of the simple example completes the development of the packing approach. Section 4.9 summarises its main features and reflects upon the required characteristics Section 4.1 outlined.

4.9 Conclusions

This Chapter has developed a parametric ship description suitable for the generation of service vessels configurations. Two simple observations in Section 4.2: firstly, ships have a finite size, and secondly, not all systems onboard may overlap, pointed towards a packing-based approach as a suitable model for ship architecture.

Following the decision to adopt packing as the basis for the parametric ship description, Section 4.4, discussed the requisites of generic packing problems, which Sections 4.4.1 to 4.4.5 subsequently tailored to handle the early stage design of service vessels.

Section 4.5 considered whether to apply one of the packing requisites: design changes, in a sequential or concurrent manner to best ensure that the large set of objects that describe the ship configuration complies with overlap rules. Next, Section 4.6 introduced the means naval architects can use to focus the ship description towards particular configurations without overtly restricting the ship description.

The resulting packing process was covered in Section 4.7; it combines the findings from the previous Sections into a coherent packing approach that converts changes in input parameter values into changes in ship configuration.

Sections 4.9.1 and 4.9.2 respectively summarise the most important features of the packing approach and considers whether the packing-based approach is suitable to serve as a parametric ship description for the early stage design of service vessels by reflecting upon the required capabilities stated in Section 4.1.

4.9.1 Summary of the most important features of the packing approach

This Section provides a summary of the most important features of the packing approach, highlighting the key decisions that enable a packing approach to generate coherent service vessel configurations.

- Objects as proposed by *Van der Nat [120]*: user-defined clusters of systems fulfilling one or more functions, describe the entire ship configuration (Section 4.4.1). In doing so objects inherent the spatial features of the systems they represent, which leads to six object types: envelope, subdivision, hard, soft, free space and connection that together describe the configuration (Section 4.4.1). A seventh object type: logical, focusses the description towards particular configurations (see below).

- Three types of overlap: prohibited, required and permissible, determine whether pairs of objects may occupy the same part of the positioning space (Section 4.4.3). The three types of overlap are distributed over pairs of objects types (Table 4.1). Logical objects have bespoke overlap rules to help focus the configuration (discussed in Sections 4.4.3 and 4.6).

- All objects are packed in a box-shaped positioning space with a size equal to or larger than the envelope to enable the packing approach to perform variations in envelope shape and size (Section 4.4.2). It also supports different ways to model interactions between interior and exterior demands on the envelope (e.g., ‘packing’ or ‘wrapping’). The latter was studied in Section 4.5.5 where a consideration of the impact of sequence led to the decision to pack the envelope as the very first object (see below).

- The packing process uses overlap management to convert the ship's initial description, i.e., the initial shape and position of objects, into a final description with altered shapes and positions. In the process, two design changes: position change and shape conformation, alter the objects' positions and shapes to comply with the overlap rules (Section 4.4.5).

  A third type of shape change, applicable to the envelope only, was discarded in Section 4.5.5 after illustrating that the particular shape change can be replaced by using a specific sequence that packs the envelope as the very first object (Sections 4.5.3 to 4.5.8).

- Input parameters from the search algorithm can change the initial position and shape of objects, as well as their presence, number and connectivity, enabling the variation of ship configurations by a search algorithm and, crucially, without human interaction. The initial shape and position are changed by overlap management to ensure compliance with overlap rules (Section 4.4.5).

  By using x, y and z-coordinates as parameters to determine the initial positions, the packing process can generate ship configurations that have a variable packing density, enabling it to cover space-driven and weight-driven ship types, describe configurations at a variable level of detail and, lastly, configurations that use excess space to improve performances, e.g., sea-keeping (Section 4.4.5).

  Note, the amount of excess space is dependent on ratio between the volume enclosed by the envelope and the volume occupied by the other objects and is thus not determined beforehand. It will therefore vary between feasible designs, with some being more spacious and others being more dense.

- Packing takes place by sequentially packing systems (= groups) of objects that undergo overlap management concurrently (see Section 4.5). The sequence of packing is user-defined (see Section 4.5.1) and does not change during the packing process or search process. Overlap management is applied only to the current system of objects; all other objects packed previously are left unchanged. This sequential approach simplifies the implementation of overlap management considerably and also helps to focus the packing process (Section 4.5.7).

  The reason to pack multiple objects concurrently is that the representation of a cluster of systems often requires several different object types. Packing these different types of objects concurrently helps the system of objects to correctly model the cluster of systems they represent (see Section 4.5.1). Overlap rules are discarded within a system of objects to simplify modelling.

  The impact of the packing sequence was considered in Sections 4.5.3 and 4.5.4. This resulted in the decision to pack the envelope as the very first object, after performing a simple test case which illustrated that packing the envelope first does not appear to restrict the search algorithm from generating a wide range of useful configurations. This decision enabled the removal of one of the two shape changes discussed in Section 4.4.5, which the envelope uses for overlap management, simplifying the implementation of the packing approach (Section 4.5.5).

  The resulting sequence proposed in Section 4.5.7 packs the envelope first, subdivision objects representing decks, second, large and constrained systems of objects (soft or hard) that may impact bulkhead positions third, subdivision objects representing bulkheads fourth, before it continues with subsequent systems (ordered roughly from large and constrained, to small and unconstrained).

- The packing process offers several ways to focus the description towards particular configurations. Among these are constraints based on restrictions to initial and final x, y and z-coordinates, the use of positioning spaces with pre-defined constraints, a particular user-defined sequence and the most flexible option: logical objects that restrict the relative positions of systems of objects.

The features discussed above form the basis for the conclusions drawn in Section 4.9.2, which establish whether the packing-based approach is suitable as a parametric ship description.

4.9.2 Is packing suitable as a basis for a parametric ship description?

To conclude this Chapter, this Section returns to the characteristics the packing approach should have, as stated in Section 4.1. The extent with which they indeed have been achieved is discussed below.

- *'Describe the whole of the ship at a level of detail deemed suitable by the naval architect'*. The six physical object types should enable the naval architect to describe the entire ship, i.e., hull, superstructure, her systems and their connections (as was illustrated by the simple application in Section 4.8). A variable level of detail is achieved in two complementary ways. First, using more, or less, objects to represent the same set of systems enables one to describe a ship at, respectively, a higher and lower level of detail. Second, using x, y and z-coordinates that vary the initial positions of systems of objects generates configurations with a variable packing density, one of the key requisites needed to handle a variable level of detail, as more detailed descriptions become less full with objects (see Section 4.4.5).

- *'Ensure reusability for various ship types'*. Chapter 6 will establish reusability more firmly by applying the packing approach to two very different service vessels, one civilian and one military. Still, two aspects form the basis for reusability. Firstly, the object types defined in Section 4.4.1 are generic enough to cover a whole range of ship types, e.g., all service vessels use an envelope and subdivision objects. Moreover, the wide range of shapes that hard and soft objects can describe should be sufficient to handle a broad range of service vessels. Secondly, handling a variable packing density enables the packing approach to cover both space-driven and weight-driven ship types and also enable the packing-based ship description to generate configurations with 'excess space' that have improved performances.

- *'Handle large changes to the ship definition introduced by changes in input parameter values'*. Section 4.4.5 illustrated how input parameters change the initial shape and position of objects. Overlap management subsequently converts these initial shapes and positions into final shapes and positions that comply with the overlap rules. A key feature is that changes in input parameters results in changes of the shape and position of individual objects. Overlap management subsequently determines how these individual changes affect the overall ship configuration.

- *'Apply these changes concurrently to the entire ship description, i.e., hull, superstructure, decks, bulkheads and systems'.* Section 4.4.5 illustrated the ability of input parameter to make large changes to the initial shape and position of objects. Subsequently, the sequential application of overlap management to all systems of objects (Sections 4.4.5 and 4.5) propagates the effect these individual changes have on other systems of objects, enabling the entire ship description to change concurrently (as illustrated by the example in Section 4.8).

  As a result, the packing process is able to produce different superstructure shapes, different deck heights and bulkhead positions, and most importantly, create different configurations of objects in the ship, solely using changes in input parameter values. This enables the search algorithm to change the entire ship configuration without human interaction, greatly increasing the speed with which ship configurations can be generated and assessed with respect to the design requirements (a crucial feature of the approach outlined in Section 3.6).

- *'Support a variable degree of focusing by the naval architect'.* Section 4.6 discussed the various ways in which the naval architect can focus the parametric ship description towards configurations deemed 'promising', e.g., to increase yield. Among the focussing methods is the seventh type of object: logical objects. It is important to note that the naval architect can use as little or as much focussing as deemed necessary. Furthermore, the focussing methods are flexible and simple to change when necessary (see discussion in Section 3.5.2).

Summarised, the comparison with the requirements laid out in Section 4.1 (that themselves followed from the discussion in Chapters 1 to 3) reveals that the packing-based approach developed in this Chapter should be suitable for use as a parametric ship description for the early stage design of service vessels.

Chapter 6 will combine it with a search algorithm and ratings from Chapter 5 and apply it in two test cases to further assess its capability and, most importantly, see if it indeed can generate the large and diverse set of alternatives required to cover the many options facing the naval architect.

4.9.3 Closure

This Chapter developed a packing-based ship description that uses input parameters from the search algorithm to generate ship configurations, as shown earlier in Fig. 4.1. Still, generating configurations is not enough and therefore, Chapter 5 will discuss the next three steps of the loop in Fig. 4.1: the search algorithm, the performance prediction tools and the ratings that guide the search process.

Chapter 5

Searching for ship configurations

5.1 Introduction

Chapter 4 discussed the development of a packing-based approach able to change the entire ship description solely by altering input parameter values. The packing approach's development is crucial, as it alone determines both the range of ship types it can describe and the range of design changes the description can undergo.

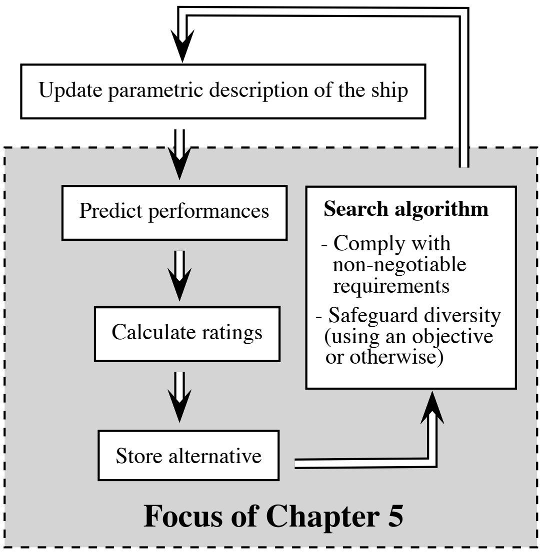


Fig. 5.1 – Focus of this Chapter

Still, the packing approach alone is not sufficient to search for ship configurations. For that, one needs a search algorithm in combination with performance prediction tools and ratings (Fig. 5.1 and Section 3.3). This Chapter therefore introduces the search algorithm used in this thesis. It also discusses the performance predictions used to evaluate the ship configurations the packing approach generates.

Most importantly, it explains how performance predictions are aggregated into ratings that enable the search algorithm to search for a set of designs that complies with non-negotiable design requirements and that is sufficiently diverse to support the investigation of a broad range of trade-offs between negotiable requirements (Fig. 5.1 and Section 3.5.3).

The Chapter is divided into five Sections. Section 5.2 introduces the search algorithm, Section 5.3 discusses the performance prediction tools used, while Section 5.4 discusses the ratings used by the search algorithm. Section 5.5 discusses integration issues that arise when combining the packing approach from Chapter 4 with the performance prediction tools, the ratings and the search algorithm introduced in this Chapter. Section 5.6 concludes this Chapter.

5.2 The search algorithm: NSGA-II

5.2.1 Selecting a search algorithm

Section 3.3 discussed the generic structure of search algorithms. A generic structure, is not enough, however. One needs an actual search algorithm to enable the approach outlined in Chapter 3 to work in practice. More specifically, one can either choose to develop a new search algorithm from scratch or to use an existing search algorithm.

Several efforts discussed in Chapter 4, i.e., at Clemson University (e.g., *Tiwari et al. [115]*) and at the University of Michigan (e.g., *Daniels and Parsons [28]*), used the first option and spent considerable effort to develop tailor-made search algorithms to better handle their packing approaches. The research presented in this dissertation differs. The author spent no time on developing bespoke search algorithms, and used an existing search algorithm, instead. The reason was simple, it provided more time for developing the packing-based ship description discussed in Chapter 4.

This, however, raises the question how one should choose an appropriate search algorithm. The selection is complicated by the many different search algorithms that exist. See *Hillier and Lieberman [58]* for an overview of the different types of search algorithms and *Rasheed and Hirsh [103]* for a discussion on the factors to consider during selection.

Five aspects were considered in the selection of the search algorithm used in the research presented in this dissertation.

1. Ability to handle multiple objectives and multiple constraints. Section 1.4 distinguished between negotiable and non-negotiable design requirements, which Section 3.3 argued can be represented by objectives and constraints, respectively. Furthermore, even though the objectives and constraints to be used are yet to be introduced in Section 5.4, it nonetheless is unlikely that only a single rating -be it objective or constraint- will be used. The search algorithm must therefore consider multiple objectives and multiple constraints without attributing them with an a-priori defined weighting (see Section 2.4.3).

2. Ability to change multiple parameters concurrently and independently. The parametric ship description from Chapter 4 uses a considerable number of input parameters to alter all parts independently. Section 3.4.2 argued that a systematic variation of input param-

eter values is not feasible, and therefore the search algorithm should be able to change multiple input parameters concurrently to speed up the search process.

3. Ability to handle complex relations between input parameters and objectives and constraints. The packing-based ship description creates a complex relation between input parameters on the one hand, and objectives and constraints on the other hand. The search algorithm should handle such complex search problems without trouble. This means, for example, that the search algorithm should be able to handle ratings that lack continuity and lack mathematically smoothness (defined as a function whose second derivative is continuous). Among the types of search algorithms discussed by *Hillier and Lieberman [58]* that can handle such complex problems are genetic algorithms, which were used extensively by the applications discussed in Chapter 2.

4. Availability. Also important is availability. The time required to implement a search algorithm can be reduced or removed altogether by reusing an implementation available in *Matlab*, the program chosen in Chapter 4 for the development of the parametric ship description (see Section 4.4.4). Some modifications may be required, but they will take less time than implementing and debugging a search algorithm from scratch. Using a search algorithm implemented in the same programming language as the packing approach also enables a tight integration, which facilitates ease of use.

5. Reputation. Using an existing, well-established search algorithm offers a good indication of its search ability. This removes the need to implement several alternative search algorithms, perform test cases on several search problems and compare the results, which makes more time available for the development of the packing approach. The reputation of algorithms used by the references discussed in Chapter 2 was considered to warrant special attention.

5.2.2 A suitable candidate: the NSGA-II genetic algorithm

Several types of algorithms meet the first three requirements outlined in Section 5.2.1, e.g., simulated annealing (*Kirkpatric et al. [68]*), genetic algorithms (*Mitchell [83]* and *Hillier and Lieberman [58]*) and particle swarm optimisation (*Kennedy et al. [65]* and *Pugh and Martinoli [101]*). Meeting the first three requirements makes the last two requirements from Section 5.2.1 more important and they are considered in more detail.

One genetic algorithm with a good reputation available to the author, is NSGA-II (see *Deb et al. [32]* for details). It has been used extensively for a wide range of applications reported in *Bernardoni et al. [15]*, *Atiquzzaman et al. [14]*, *Malyna et al. [76]*, *Pouw [100]* and *De Oliveira [30]*.

Crucially, NSGA-II was used by Clemson University in combination with their packing approach (see *Miao et al. [82]*). This means NSGA-II is suited to use initial x,y and z-positions as input parameters, a key feature of packing approach developed in Chapter 4 that is shared with Clemson's approach.

Just as important, the NSGA-II implementation *Pouw [100]* developed for shape optimisation of ship propellers was written in *Matlab* and was made available to the author. *Pouw [100]* also performed several benchmarks to ensure the implementation works correctly (see *Pouw*

[100] for details). Using *Pouw [100]*'s implementation helped reduce the effort required to implement a suitable search algorithm for use with the packing-based ship description developed in Chapter 4.

Consequently, the NSGA-II algorithm implemented in Matlab by *Pouw [100]* was chosen and has been used in the applications in Chapter 6. Note that the selection of NSGA-II is not considered to be definitive. It can be replaced readily by an improved search algorithm, when one becomes available.

Section 5.2.3 explains NSGA-II's workings in more detail.

5.2.3 Outline of the NSGA-II algorithm

NSGA-II stands for 'Non-dominated Sorting Genetic Algorithm' and falls in the broad group of search algorithms called genetic algorithms (see *Goldberg [47]* and *Mitchell [83]* for an introduction).

This class of search algorithms mimics biological evolution by 'evolving' a 'population' of designs towards better ratings over several 'generations'. The population consists of multiple individuals (an individual is also called a chromosome): a vector stores the input parameters for an individual design, while a single input parameter value is called a 'gene'. The search process used by NSGA-II to evolve the population follows the workflow outlined in Fig. 5.2. The process to generate a new population is called 'replication' and uses individuals from the 'parent' population to create a new 'child' population of individuals. Replication relies on selection, cross-over and mutation operations, which are explained below. Next, the new 'child' population is evaluated, i.e., the objective and constraint values are calculated for each individual in the population (Section 5.4 discusses the objectives and constraints used in this dissertation). Lastly, NSGA-II merges child and parent populations and sorts them to retain the 'best' individuals from both child and parent populations (using the prioritisation from Fig. 5.5 to consider objective and constraint values); individuals with insufficiently good ratings are removed from the search process (however, the resulting configurations, performances and ratings are stored). Note that individuals that do not satisfy constraints are considered infeasible, while those that do are considered feasible (following *Deb et al. [32]*).

The operations used by NSGA-II to search for improved ratings are: initialise, evaluate, select, cross-over, mutate and elitism. These operations are applied in a recursive manner, similar to the generic search algorithm discussed in Section 3.3. Furthermore, the user determines both the number of generations NSGA-II repeats the evaluate, select, cross-over, mutate and elitism operations (shown in Fig. 5.2) as well as the size of the population, i.e., the number of individuals the operations are applied to.

The operations in Fig. 5.2 are discussed in more detail below.

- Initialise. Initialisation concerns the generation of the parameter values for the first population. Obviously, the values of the parameters should match any user-defined bounds on those parameters. Moreover, the implementation developed by *Pouw [100]* uses input parameters that can handle real numbers. Main benefit of real numbers over binary numbers is their ability to handle any kind of value change. The initial values are determined at the beginning of the search process using *Matlab*'s random number generator.

```
1. Initialise population by generating random values.

2. Evaluate individuals: calculate objectives and constraints.

For current_gen = 1:number of generations

    3. Create child population using

        - Selection
            - Tournament selection
            - Choice of parents based on
                - Feasibility
                - Rank
                - Crowding distance
        - Cross-over
        - Mutation

    For 1:population size

        4. Evaluate designs (calculate numerical ratings)

    End

    5. Ensure elitism:
        - Merge parent and child population
        - Perform priority rating according to
            - Feasibility
            - Rank
            - Crowding distance
        - Create new parent population by retaining
          individuals with highest priority ratings,
          as shown in Fig. 5.5

End
```

Fig. 5.2 – Outline of the NSGA-II search algorithm

- Evaluate. Evaluate uses the parameter values to calculate the ratings, i.e., objectives and constraints, using the process shown in Fig. 3.2 and 5.1. The evaluation process is not discussed further, given the elaborate discussion of the packing approach, prediction tools and ratings in, respectively, Chapter 4, Section 5.3 and Section 5.4. The ratings that result from the calculation process are used in the selection operation.

- Selection. After evaluation, the ratings of the population of designs are used to guide the creation of a new population. To this end, individuals need to be chosen that form the basis with which to create a new population. Selection determines how designs are chosen for this purpose.

  Discussing the selection process requires an elaboration as to how NSGA-II considers multiple objectives and constraints during the search process. Objectives are considered using two measures: rank and crowding distance, while constraints are treated using the total sum over all constraint violations. Again, see Section 5.4 for the objectives and constraints used in this dissertation.

Objectives are discussed first. NSGA-II employs a Pareto-based approach to identify trade-offs between multiple objectives (see *Pareto [93]*). This approach was briefly mentioned in Sections 4.3 and 4.5. Individuals in a population are Pareto-optimal when it is impossible to choose another individual that has improved values for all objectives. Improving one objective value of a Pareto-optimal individual is possible only by deteriorating at least one other objective value. Pareto-optimal individuals are considered to 'dominate' other individuals and are themselves considered to be 'non-dominated' (hence the name of the NSGA-II).

Fig. 5.3 shows an example. It shows ten individuals with two objectives: A and B (both objectives are to be minimised). Individuals no. 1, 3 and 9 are non-dominated and located on a so-called Pareto-front: a set of non-dominated individuals. Take individual no. 9 for instance. One can only reduce the value of A by selecting individual no. 1. This does come at a cost: B increases to reduce A. The individuals along a Pareto-front offer the naval architect a range of compromises between objectives, i.e., one can choose to reduce A to improve B or viceversa after the search process. Pareto-fronts thus offer the naval architects the option to settle trade-offs a-posteriori (see Section 2.4.3).

The population of individuals can be divided into multiple Pareto-fronts. For example, the solid line shows the first Pareto-front in Fig. 5.3. Removing the individuals in the first Pareto-front (no. 1, 3 and 9) and sorting for non-domination in the reduced population results in the second Pareto-front (the dashed line in Fig. 5.3). NSGA-II uses the Pareto-front in which an individual resides as a measure called 'rank'. Lower Pareto-fronts are considered to better than higher, e.g., individuals in the first Pareto-front are considered better than those in the second and subsequent Pareto-fronts.

Using the number of the Pareto-front in which an individual is located is insufficient. One also needs to distinguish between designs within a single Pareto-front (as in Fig 5.3). NSGA-II uses the concept of 'crowding distance' to do so. Crowding distance is defined as the average side length of the box drawn between the neighbours of an individual on the Pareto-front. Fig. 5.4 shows the same set of individuals as in Fig. 5.3, including those on the first Pareto-front. The crowding distance of individual no. 9 is defined by the average dimensions of the dashed box, i.e., $(|A_3 - A_1| + |B_1 - B_3|)/2$. Crowding distance is normalised by dividing the measure by the difference between the maximum and minimum value for each objective. Purpose of crowding distance is to improve the spread of designs along the Pareto-front, which offers the naval architect a broader spectrum of compromises between A and B.

Also left unconsidered are individuals that fail to satisfy the constraints. Assume an individual indeed violates constraints. The total constraint violation is calculated by adding together the values of each individual constraint violation. Constraints that are satisfied are left out of the calculation. The result is that individuals that comply with all constraints have a total constraint violation equal to zero, while other have non-zero values.

The rank, crowding distance and total constraint violation are used in a prioritisation to help select 'better' individuals, using the scheme shown in Fig. 5.5 (reproduced from *Pouw [100]*). It illustrates how NSGA-II considers constraint violation, rank and crowding distance in a sequential manner. Selection of individuals takes place once the prioritisation of individuals has been established, using a process called tournament selection (see *Mitchell [83]* for an example).

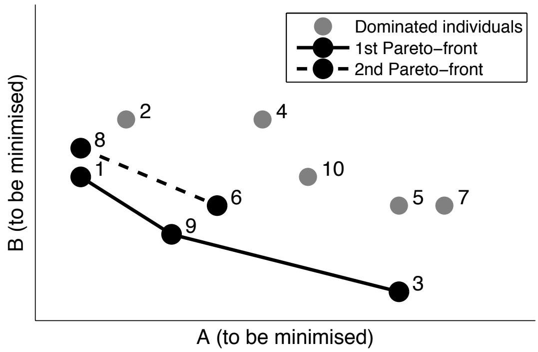


Fig. 5.3 – Set of ten designs with the first two Pareto-fronts

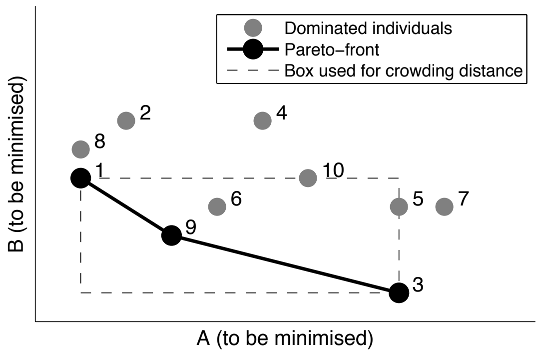


Fig. 5.4 – Box used to define crowding distance

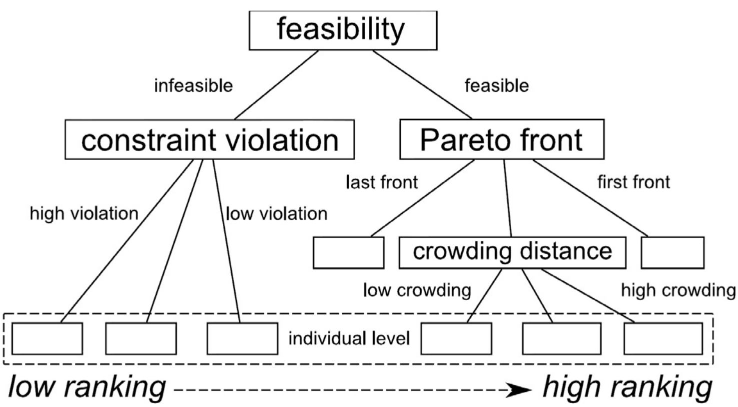


Fig. 5.5 – Prioritisation used by NSGA-II to guide the selection and search process (from *Pouw [100]*)

In this process, the algorithm chooses two pairs of individuals from the parent population in a random manner. From each pair, the individual deemed most fit is chosen for reproduction, following the prioritisation in Fig. 5.5. This process is repeated so that all individuals in the population are considered during the selection process. The resulting selection process and the prioritisation it uses guides the search process towards feasible individuals first, then towards non-dominated individuals and then to a suitable spread along the Pareto-front, once the generation of new Pareto-fronts proves impossible.

Note that selection only considers those ship characteristics for which ratings were invented (see the comment on the lack of completeness of ratings discussed in Section 3.4.2). As such, it is the selection process that may guide the search towards 'too optimal' designs (as discussed in Section 3.4.2). Still, the random selection of pairs of individuals for the cross-over operation (discussed below) and the consideration of all individuals in the parent population during selection process leaves ample scope to select 'sub-optimal' individuals as parents. This can lead, potentially, towards 'sub-optimal' children, which offer one way to address the lack of completeness and the 'too optimal' individuals that result from it (as has been argued in Section 3.5.2).

- Cross-over. The two selected individuals, i.e., 'parents' are used to generate two new individuals, i.e., 'children'. The process consists of two parts, the first of which is the cross-over operation. It mimics biological cross-over to swap genes between two chromosomes of living creatures. For the use in NSGA-II, it means that input parameter values from one parent are transferred to another (and viceversa) using a so-called 'simulated binary crossover' process (discussed in *Deb and Agrawal [31]*). Fig. 5.6 shows a simple example of a cross-over operation to illustrate the concept. The example performs the cross-over at a single random position, while simulated binary crossover used by NSGA-II considers cross-over for each pair of input parameters individually. Furthermore, simulated binary crossover may produce input parameter values not present originally in the two parents (still, the chance of this happening is quite small), which will increase the diversity in the child population. Also, whether and what parameters are exchanged during cross-over depends on a user-defined cross-over probability (see Section 5.2.4 and Table 5.1). The result of the cross-over operations are two children, each with a vector of input parameter values derived from both parents. These values are adjusted in the second operation: mutation, which is discussed below.

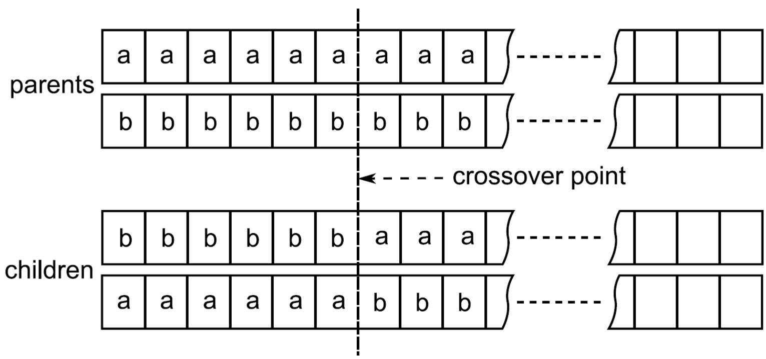


Fig. 5.6 – A simple cross-over operation that swaps parameter values between two parents to generate two children (from *Pouw [100]*)

- Mutate. The second operation is called mutation and modifies the children after the cross-over operation finishes. Again, the operation is copied from biology, where individual genes undergo random alterations. For NSGA-II, it means that values of individual input parameters are changed by replacing them with a new, randomly generated random value. For example, imagine that a mutation changes one of the parameter values in Fig. 5.6 from a into c. These value changes are generated using a similar process as the 'simulated binary cross over' operation discussed previously. Whether and what parameters are mutated depends on a user-defined mutation probability (similar to the cross-over operation, i.e., see Section 5.2.4 and Table 5.1).

  The process to generate the children from the parents is completed once the mutation operation has been carried out. The resulting children are evaluated (as discussed above), i.e., their ratings calculated, after which both parent and child populations have known ratings and can be compared. The next operation determines which individuals from each population to keep. This is discussed below.

- Elitism. The selection, cross-over and mutation operations result in a newly created child population, which the evaluation of the individuals attributes with objective and constraint values. Once created and evaluated, the child population does not replace the parent population straight away. Instead, NSGA-II applies a sorting procedure to retain the best individuals from both populations in the new parent population used in the next generation. The process first combines parent and child populations in a single, merged population. Next, the same prioritisation shown in Fig. 5.5 is used to fill the new parent population, starting with the highest ranked individuals. This new parent population is used as to generate a new child population in the subsequent generations. This provides NSGA-II with elitism, i.e., it ensures no 'good' designs from the parent population are inadvertently lost during the search process.

Together the operations enable NSGA-II to search for designs with the desired ratings. More specifically, it will search according to the prioritisation shown in Fig. 5.5, first for feasible designs that comply with all constraints, then for non-dominated designs and then for diversity along the Pareto-front.

In summary, the search ability ensures NSGA-II is suitable for use with packing-based ship descriptions. Its use of real numbers as input parameters supports the application of x,y and z-coordinates that determine an object's initial position prior to overlap management (discussed in Chapter 4), while the ability of NSGA-II to handle discrete values (by rounding parameters towards the nearest integer) facilitates the selection of systems, for instance (Section 3.5.1). Lastly, treating the number of systems that could not be packed as a constraint (see Section 4.7.3) focusses the search towards those configurations that can pack all systems of objects successfully.

The objectives and constraints used by NSGA-II are discussed in Section 5.4. Before doing so, Section 5.3 introduces the performance predictions on which they are based.

5.2.4 Overview of NSGA-II settings

The NSGA-II settings summarised[1] in Table 5.1 were used for the two applications in Chapter 6. The variation in the number of generations and number of configurations is caused by the fact that each search run used a different number of generations. Furthermore, no thorough investigation was made of the detailed impact of these settings, under the assumption that the lack of completeness, i.e., the incomplete set of ratings used to guide the search process (as discussed in Section 3.4.2), will have a far larger influence.

Setting	Values for warship application	Values for drillship application	Unit
Population size	60	120	(-)
No. of generations	393 to 881	42 to 112	(-)
No. of configurations	23580 to 52860	5040 to 11200	(-)
Cross-over probability	0.9	0.9	(-)
Mutation probability	0.9	0.9	(-)

Table 5.1 – Summary of the NSGA-II settings used for the applications in Chapter 6

5.3 Performance prediction tools

Performance prediction tools are used to evaluate the ship configurations generated by the packing approach, and, in order to use the packing-based ship description to actually design ships, one must therefore decide upon the performance prediction tools to be used. This creates a problem, since the performance predictions depend -in part- on the type of service vessel under consideration.

One of the issues outlined in Section 4.1 considered the reusability of the packing-based ship description. In order to investigate its reusability without taking the effort to develop at least two complete set of different performance prediction tools, it is desirable for the performance prediction tools to be reusable, i.e., generic, as well.

Fortunately, Section 3.5.1 argued many service vessels share the same of basic performance requirements. It is this set of performances familiar to any naval architect: ship weight and centre of gravity, hydrostatics, ballasting, intact stability and reserve buoyancy, and resistance, propulsion and endurance predictions, that will be used in this dissertation. In addition, two measures dealing with quality of configuration are also used.

The set of performance prediction tools is not intended to be exhaustive (it cannot be, due to the lack of completeness of ratings which was discussed in Section 3.4.2). Instead, the set should cover those performances dealing with the most important non-negotiable design requirements, and with those performances with the biggest influence on the ship's design. Additional performance prediction tools can be added should the need arise, such as the approach to assess warship vulnerability developed by *Van Ingen [125]*, which uses the ship configurations from the packing approach as input and *Van Bruinessen [119]* who integrated an existing sea-keeping code for operability prediction.

The performance predictions used in the applications in Chapter 6 are discussed below.

[1]This Table summarises the sixteen test cases shown in Table 6.1 and 6.8.

- **Estimate of ship weight and centre of gravity**. Chapter 4 did not discuss weight estimation or the estimation of the centre of gravity. The reason is that both estimates are considered to be performances, as both a weight estimate, and a centre of gravity estimate can be carried out only once the complete ship configuration is available. This sequentiality does not mean that the influence of both of these performances on the ship configuration is neglected wholesale. Instead, their large impact is handled after the packing process is completed by applying constraints on intact stability and trim, which ensures that the search algorithm takes the influence of ship weight and the location of the centre of gravity on the ship's design into account (as discussed in Section 5.4).

  The weight and centre of gravity estimation is performed in three steps. First, the weight and centre of gravity of the ship's structure is estimated based on a specific weight per area of deck, bulkhead and shell (for the frigate in Chapter 6) or per volume (for the drillship in Chapter 6). Second, a weight and centre of gravity is attributed for the relevant object. This may concern lump weights for hard objects (such as a gun, which has a known, discrete weight) or a density-based weight for soft objects (a sleeping quarters may be considered to weigh a particular amount of kilograms per square meter of deck area, for example). Third, variable loads, e.g, fuel or ballast water in tanks, are added to the lightship weight. Together, the steps determine total ship weight and centre of gravity at full load. Note, margins for uncertainty and future growth were not included for the applications to be presented in Chapter 6, though they should be for any practical application.

  Note, no particular effort was made to develop an accurate weight estimation sensitive to both large and small changes in ship configuration. Instead, the purpose of the method used in this dissertation is to illustrate whether the packing approach can handle the impact of shifts in weight and centre of gravity caused by changes in ship configuration.

- **Hydrostatics, intact stability and reserve buoyancy**. The ship's hydrostatic properties (displacement, location of the centre of buoyancy and the meta-centric height) are calculated for a range of draughts at zero heel and trim. The calculations are made at both fixed trim and heel (both taken to be zero for the applications to be presented in Chapter 6), to reduce computational effort (there are less combinations of draught, heel and trim to evaluate). The results, when combined with the estimate of weight and centre of gravity, enables one to determine the actual draught, the meta-centric height (GM) at this draught, and the difference between the longitudinal centres of buoyancy and gravity (once the last difference is small enough, the design trim is achieved). Note, the meta-centric height is not corrected for the free-surface effects in tanks. These three performances are combined in ratings in Section 5.4. Reserve buoyancy calculations are also made and used to constrain bulkhead positions (see Section 4.4.1 and Fig. 4.22; discussed in *Van Diessen [122]*).

- **Ballasting**. Ships operate at various loading conditions and must be able to float in stable upright condition in each of them. Commonly, changes in loading condition are accommodated through the use of ballast water which ensures the ship is sufficiently stable and correctly trimmed. Using water ballast also broadens the range of ship designs for which a suitable hydrostatic situation can be achieved. It will therefore improve the yield of the search process.

  Assessing the required amount of water ballast is carried out in three steps.

- The first step checks the envelope for space that can be used to place provisional ballast tanks. More specifically, it considers only space unoccupied by other objects and located in the double bottom and wing tanks. The provisional tanks are subdivided by the transverse bulkheads discussed in Section 4.4.1 to provide the ability to reduce trim.
- The second step determines the volume and centre of gravity of these provisional ballast tanks as a function of their filling percentage. Fig. 5.7 shows an example of such tanking sounding data.

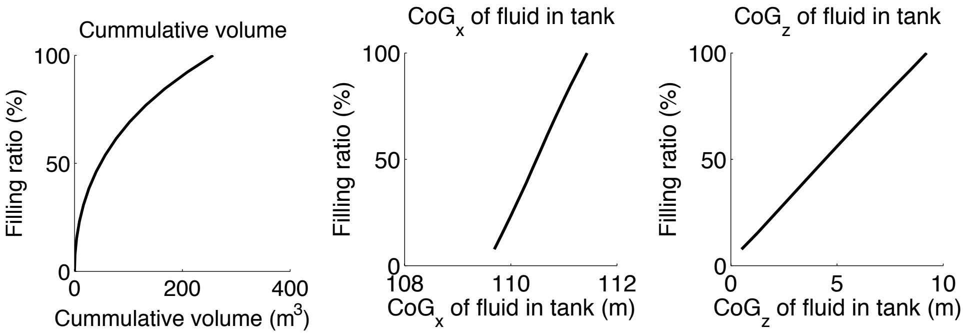


Fig. 5.7 – Example of a tank sounding table used in the ballasting calculation

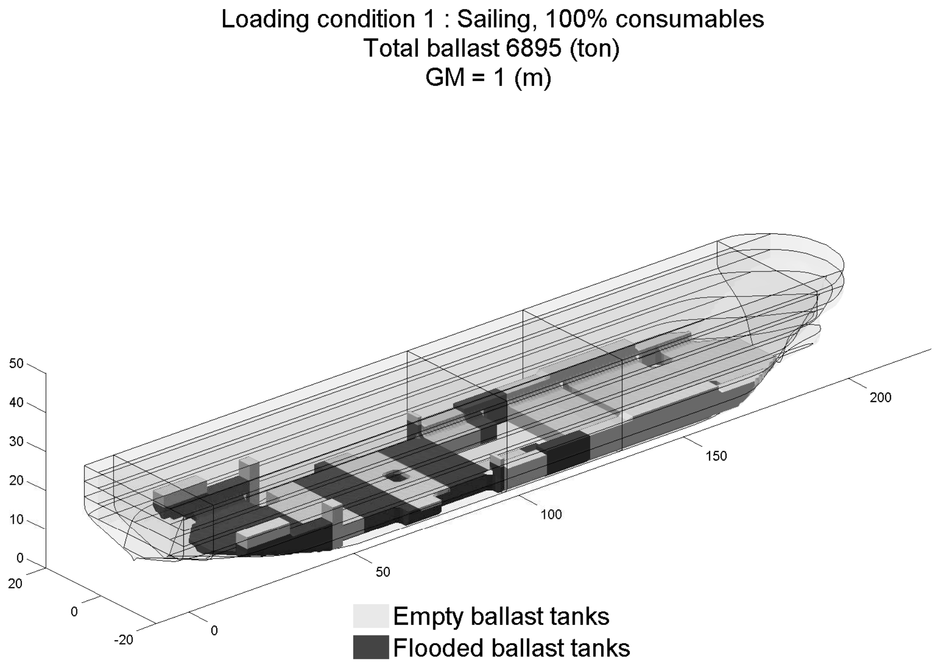
- The third step is to use a version of the NSGA-II search algorithm (described in Section 5.2) integrated in the ballasting prediction tool. It varies the filling percentage of the ballast tanks to adjust the ship's total weight and centre of gravity. NSGA-II searches for a ballast tank configuration with a minimal number of tanks and minimum tank volume that can ensure the ship is able to float upright in stable condition with zero trim and without exceeding the assumed design draught.

  Note, the ballasting calculation currently does not consider heel. The reason is that the ship configurations in Chapter 6 are largely symmetric. If required, the ballasting procedure could easily be extended to handle heel (as well as the asymmetric ballast tank arrangement it requires).

If multiple loading conditions are of interest, the third step can be repeated to check whether a suitable ballast tank arrangement can be achieved for all loading conditions (Figs. 5.8 shows an example from the drillship application in Chapter 6).

The ballasting calculation may result in ship configurations that require ballast water at full load. This situation requires the ship to drag the extra weight with her over the duration of her service life, which among other things, increases fuel consumption. Therefore, many naval architects regard this use of water ballast as undesirable, and rightly focus on designing ships that only use ballast water to compensate for changes in consumables, e.g., fuel.

The packing-based ship description is able to generate stable ship designs with limited trim that do not require water ballast (Fig. 6.37 shows one). Thus, it can help find configurations that comply with the naval architect's preference. However, there is a drawback. Not using ballast water requires considerably more attempts by the search algorithm to find feasible designs, i.e., it reduces yield considerably. For this reason, the



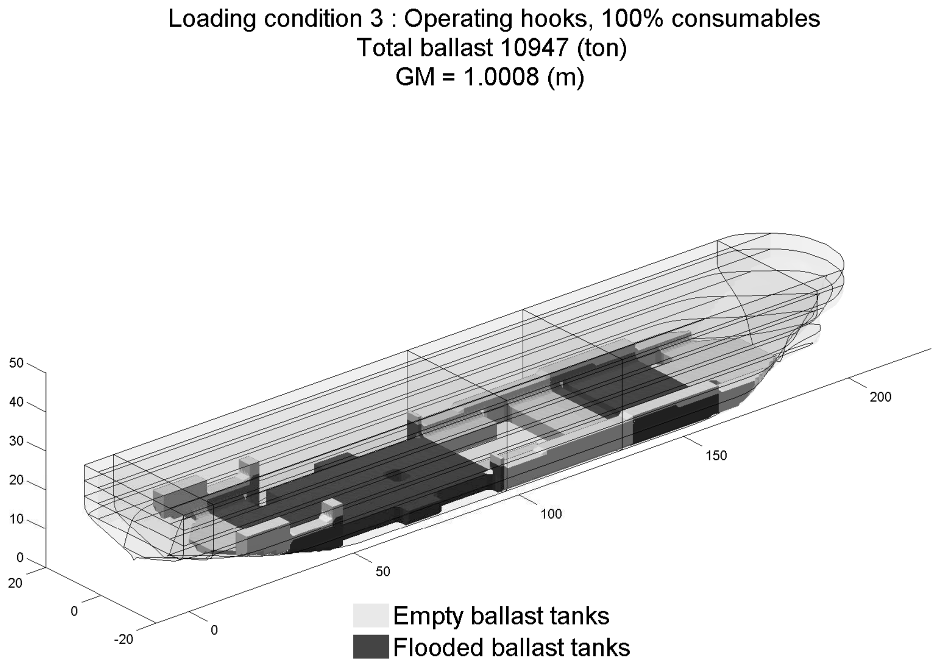


Fig. 5.8 – Ballast tank arrangement in the drillship configuration shown in Fig. 6.53, showing ballast tanks in use for two out of the eight loading conditions considered

use of ballast water is retained and has been used in the applications presented in Chapter 6, even though many configurations may use ballast water in full load condition.

Still, the increase in stability and reduction in trim the -often limited amount of- ballast water brings about in full load condition can usually also be achieved by making some other minor design changes, for example, by altering the shape of the superstructure to lower the vertical centre of gravity. This means the author considers designs with ballast water in full load undesirable in principle in the completed ship, but still useful to help find a good starting point.

In relation to this, the ballasting calculation provides a preliminary check as to whether a suitable ballast tank arrangement can be achieved. The ballasting process is not intended to determine the actual ballast tank arrangement. The actual tank arrangement depends on trade-offs between the volume of ballast water and the number of ballast tanks in the ship, as well as on requirements prescribing the ship's stability when damaged.

- **Resistance, propulsion and endurance**. One of the key performance requirements of most service vessels is the ability to attain a specified speed.

  A regression-based prediction tool (developed in-house at Defence Materiel Organisation) was used to predict the required propulsion power for the frigate application in Chapter 6. The tool is based on a regression analysis of towing-tank tests of current and past Royal Netherlands Navy surface combatants.

  First, the required propulsion power is predicted. It is used in the second step to size the propulsion plant. With the propulsion plant sized, its specific fuel consumption at cruise speed and maximum speed can be established. As such, one can calculate the required fuel tank capacity by combining fuel consumption with the required endurance at cruise speed and dash speed.

  Two remarks are in order. First, no estimate of propulsion efficiency (e.g., by making a preliminary propeller design using the systematic series in *Van Lammeren et al. [126]*) was made in the prediction tool. Instead, propulsion efficiency was assumed to be constant for the range of ships the tool can handle. Second, the propulsion power prediction uses the assumed design draught as input for the prediction. This assumption is safeguarded later on by a dedicated constraint (see Section 5.5 and also Eq. 5.3).

  The drillship application in Chapter 6 did not use a resistance and propulsion prediction. Instead, the assumption was that the installed dynamic position thrusters would be sufficient to attain the required speed (see *Wagner [135]* for details).

- **Thruster-thruster interaction**. The drillship application in Chapter 6 used a simple model to assess thruster-thruster interaction for the calculation of station-keeping ability; see *Wagner [135]* for details.

- **Packing density**. Sections 4.4.1 and 4.4.5 discussed how the use of x,y and z-coordinates as initial positions helps achieve a ship description with a variable packing density, e.g., to handle a variable level of detail and both weight-driven and space-driven designs. This choice means that packing density must be calculated to distinguish between dense and spacious designs.

  Packing density was calculated slightly different in each of the two applications in Chapter 6. The drillship application used the number of voxels inside the envelope left unoccupied by objects. The frigate application calculates packing density given by the ratio of the number of voxels occupied by objects divided by the number of voxels enclosed inside the envelope. Both measures provide the same information. However, the ratio-based measure was adopted to provide better insight when comparing sets of configurations with both different sizes and different objects (as in Chapter 6).

- **Proximity**. Several references discussed in Chapter 2 and Section 4.3, e.g., *Andrews [3]* and *Nick [88]* use proximity measures to distinguish between good and better ship configurations. Typically, this concerns measuring the distance between pairs of objects and multiplying it by a user-defined weighting (e.g., the fuzzy function used by *Nick [88]*, as in Fig. 4.14). Weights smaller than zero indicate separation is preferable, while weights larger than zero indicate proximity is preferable. The calculation is repeated for a large number of objects to capture all relevant interactions between systems in the ship. Next, the results for each pair of objects are added together to create a single rating indicating to what extent proximity and separation are achieved. This total sum rating is subsequently minimised as an objective by the search algorithm. This prediction was only used by the drillship application in Chapter 6 (see *Wagner [135]* for details).

- **Diversity of configuration**. The last prediction tool to be discussed is one that calculates how much a particular configuration differs from other configurations. The reason was outlined in Section 3.5.2: one needs a large and diverse set of ship configurations to investigate trade-offs, which, in turn, requires an estimate that expresses how diverse a particular design is in relation to a set of other designs.

  To develop the diversity measure, one must decide first which objects to consider in the calculation. The two object types that will be considered were the hard and soft objects. Reason to discard envelope and subdivision objects is that they influence the position and shape of hard and soft objects, meaning their impact can be considered indirectly. Free space objects and connection objects are similar, the former alter the shape and position of hard and soft objects, while the latter's shape and position depends on hard and soft objects they connect.

  Next issue to consider was whether to include shape, position, or both, when assessing the diversity of configuration. Hard objects cannot undergo shape changes, which leaves only their positions to be considered. Soft objects can undergo both position and shape changes. However, the impact of shape conformations of soft objects can be included indirectly, in a manner similar to the impact of the envelope. If a soft object undergoes a shape conformation, it can take a position that is not available without a shape conformation, i.e., changes in shape and position go hand in hand. This means only the positions, and not the shapes, of soft objects need consideration.

  As a result, one has to consider only the positions of hard and soft objects. More specifically, the centre of gravity of each object will be used. The positions are stored in a simple table (such as Table 5.2) that lists the centre of gravity of all hard and soft objects, for each ship configuration. Note, the coordinates of the centres of gravity are made dimensionless by dividing them by the ship's main dimensions (the length over all, maximum width and height between keel and the top of the superstructure).

  The question arises as to how to compare a table, such as Table 5.2, to similar tables of other configurations. The starting point is a simple observation: if two tables with object positions contain exactly the same values, the two configurations are the same. If a value differs, the configurations differ. If more values differ, so do the configurations, and, similarly, when values differ more so do the configurations. Generating configurations is pointless if configurations are similar and therefore diversity will be assessed by comparing individual configurations against all other configurations found so far that comply with the constraints (discussed in Section 3.5.3), as to avoid duplicating them.

This means one has to build a library of all feasible configurations found so far and update it continuously every time the search process generates more feasible configurations. The library and the diversity rating based on it provide the search algorithm with a ‘moving target’. For example, a configuration that was very diverse early on in the search process may become less diverse later on, if the search process generates numerous configurations that are rather similar to it.

The measure of diversity is based on a simple mathematical approach. For the calculation, the configuration is represented as a point in a multi-dimensional space. For example, if a configuration contains five objects with x, y and z-coordinates, it is considered a point in a fifteen-dimensional space.

Next, the Euclidian distance between the configuration of interest and all other configurations is calculated between different points in this multi-dimensional space. This results in a symmetric matrix containing the distances between all configurations. From this matrix, one must use one or more distances to arrive at a diversity measure. Using a single distance between two configurations is insufficient. It will have the same value for every pair of configurations, making it impossible to identify the more diverse of the two configurations.

Instead, a sum of the distances between a configuration and the two most similar configurations has been used to address this problem. Summing over all distances is also possible, but has been discarded because of the use of a ‘nearest-neighbour-search’ algorithm to calculate the distances (see *Cao [26]* for the implementation): the computational effort of the algorithm increases considerably when calculating all distances between all configurations.

The resulting measure: the sum of the Euclidian distances between a configuration and the two configurations most similar to it is used as a measure to indicate how much a particular configuration differs from the other configurations it is compared against. If the measure increases so does the difference between configurations, but the actual numerical value has no practical meaning. However, it does help to illustrate how diversity changes during the search process (this is discussed further in Chapter 6).

A simple example helps to illustrate the calculation. Fig. 5.2 lists seven points with two coordinates each. Fig. 5.9 plots the seven points in a two-dimensional coordinate system, while the Euclidian distances between each pair of points are shown in Table 5.3. Calculating the rating involves finding the two nearest neighbours for each point and adding both distances together. This shown in Table 5.4.

Closer consideration of both Fig. 5.9 and Table 5.3 reveals that the two closest neighbours of point number 1 indeed are points 4 and 6 (shown in Fig. 5.9). Similarly, the two closest neighbours of point number 5 are points number 3 and 7, though the distances are far larger than for point number 1. As a result, the least diverse point becomes point number 1, while the most diverse point is point number 5 (Table 5.4). The application of ratings to actual three-dimensional ship configurations is exactly the same, only the distances are calculated between points defined in a multi-dimensional coordinate system.

Point no.	x (-)	y (-)
1	0	2
2	6	1
3	4	2
4	0	0
5	8	9
6	2	3
7	1	5

Table 5.2 – Seven points used for the diversity example

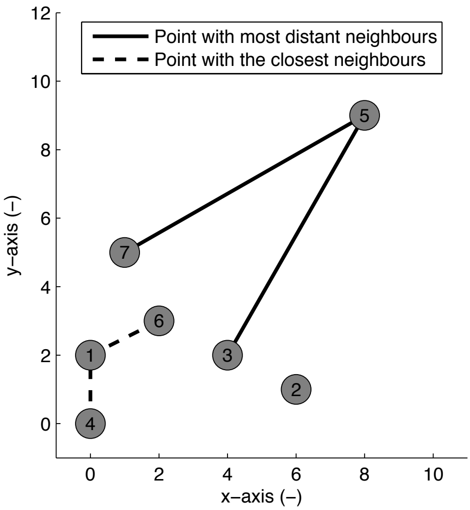


Fig. 5.9 – Points from Table 5.2 plotted

Point no.	1	2	3	4	5	6	7
1	0	6.082763	4	2	10.63015	2.236068	3.162278
2	6.082763	0	2.236068	6.082763	8.246211	4.472136	6.403124
3	4	2.236068	0	4.472136	8.062258	2.236068	4.242641
4	2	6.082763	4.472136	0	12.04159	3.605551	5.09902
5	10.63015	8.246211	8.062258	12.04159	0	8.485281	8.062258
6	2.236068	4.472136	2.236068	3.605551	8.485281	0	2.236068
7	3.162278	6.403124	4.242641	5.09902	8.062258	2.236068	0

Table 5.3 – Distances between pairs of points

Point no.	Neighbour 1	Distance (-) D1	Neighbour 2	Distance (-) D2	Diversity measure D1 + D2 (higher is better)
1	4	2.00	6	2.24	4.24
2	3	2.24	6	4.47	6.71
3	2	2.24	6	2.24	4.47
4	1	2.00	6	3.61	5.61
5	3	8.06	7	8.06	16.12
6	1	2.24	3	2.24	4.47
7	6	2.24	1	3.16	5.40

Table 5.4 – Diversity measures for each point in Table 5.2 and Fig. 5.9

Together, the prediction tools discussed above help evaluate the ship configurations that the packing approach generates. Part of the information will be used in ratings discussed in Section 5.4. The remainder is stored for application in the selection approach introduced in Chapter 7.

5.4 Ratings

The performance predictions in Section 5.3 offer an abundance of information on the behaviour of a configuration. Search algorithms cannot make use of all this information and, instead, require a summary in the form of mathematical ratings. This Section therefore discusses the ratings used by NSGA-II to direct the search process. Starting point for the discussion is Section 5.4.1 which distinguishes between negotiable requirements to be used as objectives, and non-negotiable requirements to be used as constraints.

5.4.1 Negotiability of design requirements

So far the discussion of search algorithms merely mentioned objectives and constraints. However, for the applications in Chapter 6, one has to consider two things. Firstly, which performances to use in the calculation of ratings and, secondly, the negotiability of design requirements. Both are discussed below, starting with performances with non-negotiable requirements.

Non-negotiable requirements

The following set of performances are treated as non-negotiable. They cover the basic naval architectural performances frequently considered during a feasibility assessment (an assessment which may -in practice- include additional performances, e.g., structural integrity and damage stability).

1. Sufficient space. This concerns first and foremost compliance with overlap rules from Chapter 4. This ensures all systems of objects are packed successfully so that the ship is sufficiently big enough to accommodate all systems of objects such that they fit and work correctly.

2. Buoyancy. The ship should float, i.e., have enough buoyancy to carry her weight without exceeding design draught. This also helps to ensure that the propulsion power and reserve buoyancy predictions (which both rely on an assumed design draught) are not invalidated.

3. Stability. The ship should float upright in one or more user-defined loading conditions, i.e., have sufficient initial stability by having a meta-centric height exceeding a user-defined minimum.

4. Trim. The ship's trim should be close to the design trim, i.e., the longitudinal difference between the centre of buoyancy and gravity should fall within user-defined bounds (note, design trim is taken to be zero in both applications presented in Chapter 6).

5. Reserve buoyancy. The ship should have sufficient reserve buoyancy to survive a user-defined damage length; a requirement considered when placing bulkheads during the packing process. As such, a bulkhead configuration with insufficient reserve buoyancy will cause the packing process to fail; a failure that can be detected by the non-negotiable requirements that all objects can be packed successfully (defined as that all objects have a position and shape that complies with the overlap rules).

6. Speed & endurance. Speed and endurance are treated as non-negotiable, but not used as constraints. Instead, the required power and fuel capacity is estimated at design draught and trim, and used to size the propulsion plant and fuel tanks. This translates the non-negotiable speed and endurance requirements into a sufficient space and buoyancy problem, which were discussed above.

Note that the above requirements all contain thresholds that either must or should not be exceeded. They are modelled using inequality constraints in Section 5.4.2.

Issues related to negotiable requirements

Section 3.5.2 argued that a large and diverse set of configurations provides the naval architect with a range of different trade-offs between negotiable requirements. This observation was used to avoid formulating ratings that pre-define compromises between negotiable requirements (Section 3.5.2). Instead, the selection approach with which humans can investigate and select different compromises based on engineering judgement will resolve any ratings left unformulated. As a result, ensuring diversity reduces the number of objectives considerably.

However, two questions remain. Firstly, how to achieve such diversity during the search process, and, secondly, whether there are other objectives to consider, aside from diversity. Both questions are answered below, starting with the former one.

- **Diversity as an objective?** The obvious solution to using improving diversity is to use the measure proposed in Section 5.3 as an objective to guide the search process.

  However, using diversity as an objective has an important disadvantage, which occurs when the objectives differ in their speed of improvement when the concept of Pareto-optimality is used, as NSGA-II does (discussed in Section 4.5). The reason is the concept of Pareto-optimality as used by NSGA-II allows one objective to deteriorate to

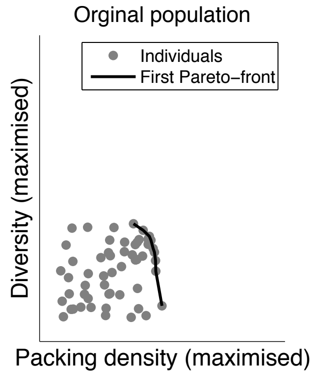


Fig. 5.10 – Original Pareto-front used as a starting point

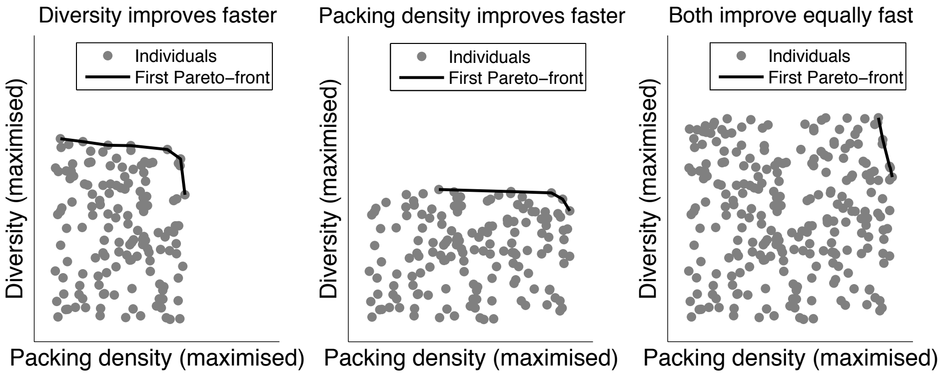


Fig. 5.11 – Pareto-fronts that result from differences in relative speed of improvement of the two objectives

improve another, but it does not actually consider during the search process which of the objectives improves and which are deteriorated.

A simple example is shown in Figs. 5.10 and 5.11. It uses two assumptions. First, diversity and packing density are used as the two objectives (the latter used for arguments given below). Second, the search process is indeed able to improve both diversity and packing density.

With these assumptions, one can consider how potential differences in the speed of improvement between the two objectives will influence the search process. The speed of improvement is measured as the average change in an objective value between subsequent generations. The image in Fig. 5.10 shows the Pareto-front used as a starting point. The three other images in Fig. 5.11 each show a Pareto-front that could develop by continuing the search effort; each results from a different relative speed with which the two objectives improve.

Though it is difficult to establish beforehand how objectives will improve exactly, it is nonetheless possible to predict the consequences of consistent differences in the speed of improvement between the two objectives. These differences will determine whether diversity is suitable as an objective.

The three Pareto-fronts resulting from the different speeds of improvement are discussed below.

– **Diversity improves faster than packing density** (left image in Fig. 5.11). In this case, the search effort focusses on improving diversity, at the cost of the improvement in packing density. This means that the configurations will predominantly differ, but will be rather spacious, perhaps to the point of being of little value.

– **Packing density improves faster than diversity** (middle image in Fig. 5.11). The opposite results in dense designs, that perhaps differ less.

– **Packing density and diversity improve equally fast** (right image in Fig. 5.11). Ideally, both packing density and diversity will improve at roughly equal speed, leading to more, and more dense, configurations that differ considerably.

Of the three options, the first: diversity improves faster than packing density, is the most likely to occur. The reason is that any difference in position between configurations will improve diversity. On the other hand, improving packing density from one configuration requires specific changes. Consequently, during the search process, diversity will -almost always- improve faster than packing density.

This is unfortunate, as it means NSGA-II will converge towards ever more diverse and -crucially- ever more spacious configurations, configurations so spacious that they are likely to be of little value to the naval architect. As a result, it means diversity is unsuitable as an objective in a multi-objective Pareto-based guided search. Therefore, a different approach is required.

Fortunately, there is another way to maintain diversity. Section 5.2.3 explained that NSGA-II uses random values during the cross-over and mutation operations. These random values can be exploited to maintain diversity and, enable the use of a single objective: packing density, to guide the search process. This ensures the child population differs so much that diversity will be maintained, while the search process focusses entirely on improving packing density. Less dense configurations will also develop during the search process and can be considered during the selection process (Section 3.5.2 discussed the need to consider all feasible designs found during the search process, regardless of their optimality, to address the lack of completeness of ratings).

• **Additional objectives.** The second question deals with additional objectives. At first sight, this looks unnecessary, as the use of diversity as an objective should be sufficient to cover all variations relevant for a broad range of compromises. Still, diversity is not used as an objective (as discussed above), requiring the definition of at least one other objective for NSGA-II to use, which will be packing density.

The reason to use packing density is that the packing approach generates configurations with a variable packing density, e.g., to improve performances and to handle a variable level of detail. Still, less dense configurations are usually of less interest[2] than more dense configurations, a preference that can be modelled by using packing density as an objective. Diversity will still be maintained thanks to the random effects in the cross-over and mutation operations, but the search process will yield an increased number of dense designs.

[2]Within limits; for example, a slightly less dense ship design might be cheaper to produce than a very dense design.

No objectives other than packing density were used, under the assumption that diversity alone will ensure a sufficiently broad range of configurations that cover a wide range of trade-offs.

Negotiable requirements

The discussion of the formulation of objectives resulted in the decision to use a single objective: packing density, as listed below.

- Packing density. Packing density is to be maximised to ensure the search process results in compact designs. Section 5.4.3 introduces the mathematical formulation of the packing density objective.

Diversity is maintained by the random effects in the cross-over and mutation operations used by NSGA-II (Section 5.2.3). As such, no objective is used to increase diversity. However, the diversity measure introduced in Section 5.3 will be used in Chapter 6. The purpose is to study how diversity changes during the search process and to see whether the reliance on the cross-over and mutation operations has any inadvertent effects.

All other ratings considered negotiable are left unformulated and are only used after the search process, i.e., in the selection approach which is developed in Chapter 7 (see Section 3.5.2).

With the discussion of negotiability completed, Sections 5.4.2 and 5.4.3 introduce the constraints and objectives to be used by the applications in Chapter 6.

5.4.2 Constraints: ratings for non-negotiable design requirements

The following constraints are used by NSGA-II.

- **All systems of objects can be packed**. The packing process stops if a system of objects cannot be packed. The number of the system that could not be packed is used subsequently to determine the total number of systems that could not be packed. This provides the first constraint g_1 shown in Eq. 5.1. Equality with zero indicates all systems could be packed successfully.

$$g_1 : \text{number of systems placed} - \text{total number of systems} \geq 0 \tag{5.1}$$

- **Sufficient reserve buoyancy at design draught**. Section 4.4.1 discussed the approach to pack bulkheads whilst ensuring sufficient reserve buoyancy and lack of prohibited overlap with prior objects. If bulkheads cannot be placed successfully, the total lack of reserve buoyancy is determined by adding the lack of reserve buoyancy for each damage case together. Damages that do not cause excessive flooding are neglected. This results in the second constraint g_2 listed in Eq. 5.2.

$$g_2 : \sum_{i=1}^{i=\text{number of damage cases}} (\text{length of excessive flooding for damage case } i) \geq 0 \tag{5.2}$$

One could argue that the reserve buoyancy criteria could be combined with the first constraint g_1 by treating the bulkheads as just any other system of objects that either fits, or not. Still, using a separate constraint, such as g_2 enables the search algorithm to distinguish bulkhead configurations that are almost feasible from those that are nowhere near feasible, speeding up the convergence of the search process.

- **Sufficient buoyancy at design draught**. The actual draught required to give sufficient buoyancy to carry the ship's weight (ballast water included, if relevant), at a specific loading condition should not exceed the hull's design draught used for the reserve buoyancy and resistance predictions. If it does, the excess draught is used as a constraint, which leads to the definition of constraint g_3.

$$g_3 : \text{assumed draught} - \text{actual draught} \geq 0 \tag{5.3}$$

- **Sufficient initial stability**. One can assess a ship's initial stability using the meta-centric height (GM). Calculated using Eq. 5.4, it should exceed a required minimum value in order for the ship float stable in upright condition.

  Sufficient meta-centric height is only a first step in achieving a stable ship design. Nonetheless, it adheres to the 'not wrong' approach outlined in Section 3.5.2, i.e., sufficient GM is a necessary but insufficient condition for a stable and safe ship.

$$g_4 : \text{GM at actual draught and trim} - \text{GM required} \geq 0 \tag{5.4}$$

- **Limited trim**. The ship is assumed to sail at a fixed design trim in all loading conditions (assumed to be zero in this thesis). As such, the longitudinal centre of gravity and the longitudinal centre of buoyancy must lie within a specific distance of each other. Ideally, this distance should be zero. However, this will likely prove too stringent for NSGA-II and have thus a negative effect on the number of feasible designs found. Instead, a more loose, inequality constraint will be used, shown in Eq. 5.5.

$$g_5 : |\,\text{LCG} - \text{ LCB at actual draught and trim}| \leq \text{ user-defined distance} \tag{5.5}$$

  The use of a zero trim constraint also simplifies hydrostatics calculation considerably. Instead of covering a wide range of trim angles and draughts, one has to calculate the hydrostatics for a limited range of draughts only, which saves considerable computational effort.

All constraints are regarded to be satisfied when their numerical value equals or exceeds zero. Negative values are regarded as constraint violations and treated accordingly by NSGA-II (see Section 5.2.3 and Fig. 5.5).

5.4.3 Objectives: ratings for negotiable design requirements

A single objective will be used by NSGA-II, following the discussion above.

- **Packing density**. The search process should focus on generating relatively dense configurations and the means to do so is the packing density objective defined in Eq. 5.6. It ensures the search process focuses on more compact designs, whilst the random effects that ensure diversity will also generate slightly more spacious designs.

$$g_6 : V_{\text{occupied by objects}} / V_{\text{enclosed by the envelope}} (\text{ which is to be maximised}) \tag{5.6}$$

Note that the NSGA-II implementation requires at least two objectives. Therefore, a constant was provided as a second, artificial objective to enable the search process to proceed without influencing its focus.

5.5 Integration issues

The previous Sections discussed the NSGA-II search algorithm as well as the performance predictions, objectives and constraints used to guide the search process. What remains to be discussed are the integration issues encountered when combining the packing-based ship description from Chapter 4 with the NSGA-II implementation developed by *Pouw [100]*.

This Section covers four of these integration issues that, when resolved, help facilitate variations, reduce computational effort, increase yield, and enable the retrieval of results. The integration itself was shown already in Section 3.6 and most recently in Fig. 5.1 and warrants not further discussion.

- **Determining the number of input parameters**. NSGA-II needs to know the number of input parameters (and their bounds) that is required to change the parametric ship description before it can start the search process. Though simple at first sight, determining the required number of input parameters becomes cumbersome when it changes frequently, e.g., by removing or adding systems of objects.

  Two improvements are used to simplify changing the number of parameters and / or their bounds.

  - First, one can run the definition part of Fig. 4.80 without the search algorithm to automatically count the number of input parameters required by each system of objects. Adding them together results in the total number of input parameters needed by NSGA-II. It relieves the naval architect from the need to manually determine the number of input parameters that are required.
  - Second, NSGA-II only generates input parameters with values between 0 and 1. When used by the relevant system of objects, these normalised parameter values are scaled according to user-defined bounds, to generate the values needed to change the parametric ship description.

    For example, an input parameter with a value of 0.5 could result in an envelope length of 100 m (if the lower and upper bounds for the length are 75 m and 125 m, respectively), or an envelope width of 20 m (if the lower and upper bounds for the width are 15 m and 25 m). This approach was first used by *Wagner [135]*).

  Together, the two improvements enable the naval architect to change the number and bounds of parameters used by the parametric ship description without cumbersome alterations to the integration of the packing approach with NSGA-II. This facilitates making variations to the parametric ship description.

- **Staggered performance calculations**. Predicting performances require both computational effort and valid input. To reduce computational effort and to ensure valid input, the performance predictions are performed in a staggered manner. For example, packing all objects successfully is used as a prerequisite for the estimation of the ship's weight, otherwise an erroneous estimate would result. Similarly, if a simple check reveals that the intact buoyancy is insufficient to carry the ship's weight, there is little use in calculating its initial stability, as the ship would sink regardless of her stability. Artificial values are supplied for those performances not predicted. These values in turn end up as constraints and direct the search effort towards configurations that do have all systems of objects packed in compliance with overlap rules, and have all performances predicted. Most importantly, the staggered calculation of performances saves considerable computational effort.

- **Storage**. All input parameter values, objectives and constraints, configurations and performance prediction results are stored for use by the selection approach presented in Chapter 7. This is done for all configurations for which all systems of objects could be packed and, hence for which performances could be predicted (any other constraint violations or objective values were disregarded). Reasons to store all designs for which all performances were predicted have been outlined in Sections 3.5.2 and 3.5.3. The most important one is the need to rapidly retrieve results to be able to study them without the need to rerun the entire packing process.

- **Use of assumptions**. The role of assumptions in the packing process has not been discussed so far, but it will be familiar to any naval architect. The reason to use them is to better handle the fact that the characteristics of systems (e.g., their size or position) can depend on the total ship configuration (see Section 1.6).

  For example, the propulsion power required to attain a particular speed depends, among other things, on the ship's displacement. Displacement depends -in part- on the weight of the propulsion system and, closing the loop, the weight of the propulsion system depends on its required propulsion power. Hence, the required propulsion power is implicit, i.e., dependent the required propulsion power.

  An extra input parameter can help to handle the dependencies in the case above. This extra input parameter varies the installed propulsion power and thus size and weight of the propulsion system. Once sized using this extra parameter, the propulsion system and other objects can be packed. If packing is successful, one can check whether the speed requirement can be met for the actual draught, using the available propulsion power. If the speed requirement is not met, the configuration is regarded as infeasible and is discarded, which leaves NSGA-II to adjust the input parameter value and try again. Though this process will yield designs that reach the required speed eventually, it is not very efficient, as NSGA-II varies a large number of other input parameters as well as the single parameter governing installed power.

  A potentially more efficient option is to use assumptions in combination with sizing or positioning rules. In that case, one can use an assumed design draught (and the lack of trim resulting from the ballasting procedure) to correctly size the propulsion plant based on the required power for the assumed displacement (the sizing uses rules). Then, the packing process can pack the objects representing the propulsion plant and all other systems of objects. If the packing process is successful, the propulsion system should have enough power to reach the required speed, provided that the total weight of the ship is equal or lower than the displacement at the assumed design draught.

As such, the problem of attaining a specific speed is translated -via assumptions and sizing rules- into an available space and available buoyancy problem, both of which are already addressed by the constraints in Section 5.4.2. Furthermore, the number of parameters reduces (no need to use a parameter to vary propulsion power), which simplifies the search process.

Using an assumption should become even more efficient when it can be reused for multiple sizing or positioning problems. For example, the assumed draught can also help the positioning of watertight bulkheads. Section 4.4.5 explained the bulkheads are placed while taking reserve buoyancy requirements into account. The actual reserve buoyancy depends on the actual freeboard and, hence, the actual draught of the ship. By predicting the available reserve buoyancy at the assumed design draught, one can place bulkheads so that the resulting reserve buoyancy suffices for the required damage length (under the assumption that the design draught is not exceeded). This is the same assumption safeguarded already by the constraints in Section 5.4.2.

The design draught is safeguarded by an inequality constraint (see Section 5.4.2). Therefore, ship designs could be found that have actual draught considerably smaller than the allowed design draught. This is not regarded to be a problem for reserve buoyancy estimate, as a larger freeboard results in more reserve buoyancy. It is more troublesome for resistance and propulsion predictions, but, in general, a smaller draught should indeed reduce propulsion power (for mono-hulls, within limits and without detailed consideration on the negative effect of a smaller propeller diameter has on propulsive efficiency). This makes the consequences of a smaller draught manageable (propulsion system should reduce in size when considering actual draught instead of assumed draught; hence, the difference in draught cannot cause an increase in ship size). Deviations between assumed and actual draught are not acceptable for the stability prediction. Stability is, therefore, predicted at the actual draught during the ballasting procedure.

However, the alternative: adjusting assumed value and actual value in several iterative loops in order to achieve equality, is considered less useful. The reason is that one may need differences between assumed and actual values (e.g., actual and assumed draught) to address the lack of completeness of ratings (see Sections 3.5.2 and 3.5.3). For example, one may need to use the difference between the actual draught and the design draught to ensure the design has adequate performance levels for performances not predicted during the search process, such as damage stability (one performance that may profit from extra freeboard). Thus, the use of inequality constraints to safeguard constraints allows one to generate designs that retain their ‘margin for improvement’.

In summary, the use of assumptions and their safeguarding by the constraints in Section 5.4.2 should enable the packing approach to estimate the size or position of a system of objects more accurately. Moreover, it simplifies the search problem NSGA-II has to tackle by reducing the number of constraints and input parameters. For both reasons, the decision was taken to implement use of assumptions in the packing approach in order to benefit from the expected increase in yield.

With the discussion of the four integration issues completed, Section 5.6 will conclude this Chapter.

5.6 Closure

This Chapter introduced the search algorithm, performance prediction tools and ratings used in combination with the parametric ship description to generate a diverse set of ship configurations that meet a basic set of non-negotiable design requirements (the grey block in Fig. 5.12).

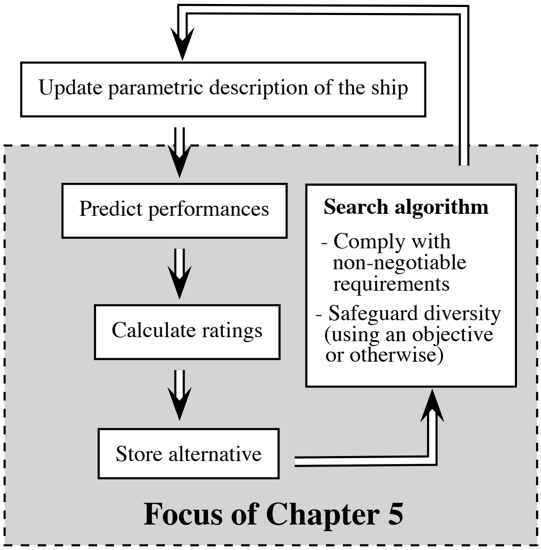


Fig. 5.12 – Focus of this Chapter

The main developments from this Chapter are summarised below.

- The search algorithm chosen to perform the search is the NSGA-II genetic algorithm (Section 5.2). It can handle large numbers of discrete and continuous parameters, complex ratings and was already used in packing applications (see *Miao et al. [82]*). In addition, the availability of a Matlab-based NSGA-II implementation from *Pouw [100]* reduced the effort required to program the search algorithm and made more time available for the development of the packing approach (Chapter 4) and selection approach (Chapter 7).
- Performance predictions. Section 5.3 provides an overview of the performances predicted. They include the basic naval architectural performances, e.g., buoyancy in intact and damaged condition, initial stability, as well as propulsion power and speed. In addition, several packing specific performances such as the violation of overlap rules, the number of systems of objects that could be packed, diversity and packing density are predicted to help guide the search process via ratings. The set of performance predictions can be expanded, if necessary, e.g., with *Van Ingen [125]*, who developed a vulnerability prediction tool for early stage warship design.
- Objectives and constraints. The performance predictions are stored for use in the selection approach (Chapter 7). Part of them are summarised in ratings to guide the search process towards particular designs with two key features.

Firstly, the configurations must comply with a basic set of non-negotiable design requirements. This is achieved by formulating basic feasibility requirements, such as sufficient buoyancy and initial stability, as constraints for NSGA-II (Section 5.4). By being relieved from the burden of complying with these feasibility criteria, the naval architect can focus on trade-offs between negotiable requirements (Section 3.5).

Secondly, the set of configurations generated by the search process must be diverse to support the investigation of different trade-offs (Section 3.5.2). Moreover, they should be dense in order to be useful for the naval architect. However, only a single objective is used: packing density. Diversity was not used as an objective, following the argument in Section 5.4.1. Diversity will be maintained, instead, by using the randomness in parameter values introduced by the cross-over and mutation operations used by NSGA-II (Section 5.2.3).

Together, the search algorithm, performance predictions and ratings in combination with the packing approach from Chapter 4 enable the generation of a large and diverse set of configurations that meet a basic set of non-negotiable design requirements. It thus complies with the overall approach outlined in Section 3.6.

With the discussion of the search algorithm, performance prediction tools, the ratings and their integration according to Fig. 5.12 complete, it is now possible to apply them in combination with the packing-based ship description. Chapter 6 presents the two applications: a drillship and a frigate, performed to investigate their ability to generate a large and diverse set of alternatives to cover the many options facing the naval architect. The resulting collection of designs will be used in Chapter 7 to select a small number of ‘promising’ alternatives.

Chapter 6

Applications

6.1 Introduction

Chapter 5 discussed the search algorithm, performances predictions and ratings which are used in combination with the packing approach from Chapter 4. Together, they should be able to generate a large and diverse set of ship configurations that meet a basic set of non-negotiable design requirements.

This Chapter investigates the ability of the resulting approach to actually find such configurations, and, as important, to see if it is able to investigate some of the options naval architects face during the early stage design of service vessels (discussed in Section 1.3). To this end, the required characteristics of the approach are repeated below; they are gathered from Sections 3.5.1 to 3.5.3.

The first set of five issues to be considered deal with the capabilities of the packing-based ship description when used with NSGA-II in an actual application. They were listed in Section 4.1.

- Issue 1: *Describe the entire ship configuration at variable level of detail, one deemed suitable by the naval architect.*
- Issue 2: *Instigate large changes to the ship configuration by changing parameter values.*
- Issue 3: *Apply these changes concurrently to the entire ship configuration, i.e., hull, superstructure, decks, bulkheads, objects and connections.*
- Issue 4: *Allow variable degree of focusing by the naval architect.*
- Issue 5: *To study whether the same packing approach can be used for different ship types.*

The second set of issues concern the ability of the packing approach combined with the search algorithm to handle a practical early stage ship design application.

- Issue 6: *Use the approach to generate a large and diverse set of feasible ship configurations.*
- Issue 7: *Assess whether the use of assumptions outlined in Section 5.5 ensures the correct speed, range and reserve buoyancy.*
- Issue 8: *Assess whether the approach is able to investigate changes in requirements that can reasonably be expected during the early stage design of service vessels.*

Two applications were carried out to investigate the eight issues. The first application deals with a frigate-type warship and is presented in Section 6.2 (a summary of the results in Section 6.2 was presented in *Van Oers et al. [133]*). Since it is the most elaborate application it will be discussed first, even though it was the second application to be carried out time-wise. Fig. 6.1 shows two of the configurations generated.

The second application is a deep water drilling-vessel introduced in Section 6.3. This application was performed as a MSc. research project carried out under supervision of the author of this dissertation. Its results are reported in *Wagner [135]* and summarised in *Wagner et al. [136]* and *Wagner et al. [137]*. The project was carried out at the ship and offshore design and engineering company *GustoMSC* in Schiedam, The Netherlands.

Note that the drillship application was carried out first. However, it was less elaborate and therefore will be discussed second. Moreover, the discussion of the drillship application limits itself to investigating those issues that were not considered in the frigate application. Fig. 6.2 shows two of the configurations generated.

It is important to note that the discussion of the results in Sections 6.2 and 6.3 focusses predominantly on investigating the eight issues outlined above. Though the naval architectural features of the designs will be covered, this Chapter does not aim to provide a full and thorough design review of each feasible ship configuration that is presented.

Also, the applications and their results should be treated as ‘proofs-of-principle’ that can be improved upon (for instance, by adding more systems of objects representing those systems currently missing in the parametric ship description). Nonetheless, the author deems them sufficiently representative to be able to investigate the issues outlined above.

Section 6.4 presents a more in-depth investigation of some theoretical issues that arose during the application of the packing-based ship description. Finally, Section 6.5 uses the results from Sections 6.2, 6.3 and 6.4 to reflect upon the eight issues in order to conclude this Chapter.

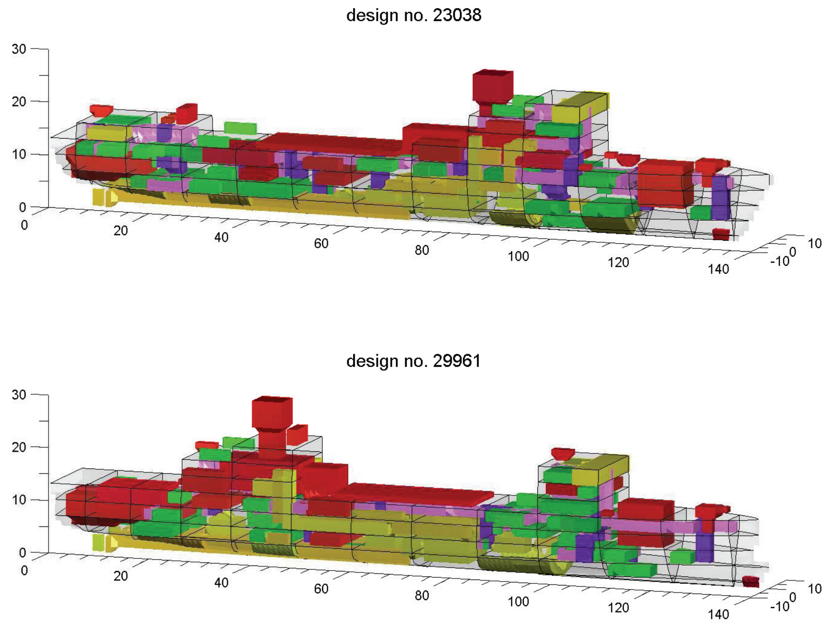


Fig. 6.1 – Two feasible frigate configurations to be outlined in Section 6.2

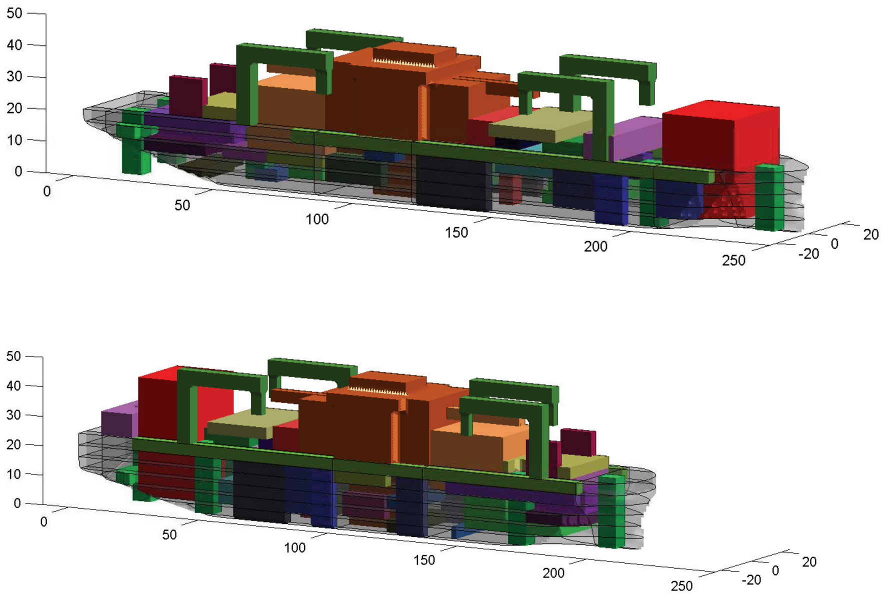


Fig. 6.2 – Two feasible drillship configurations to be outlined in Section 6.3

6.2 Frigate application

6.2.1 Overview

The frigate application looks into the applicability of the packing approach for early stage warship design as carried out at the Defence Materiel Organisation in The Hague.

Eleven search runs covering a range of variations were carried out to help investigate the issues laid out in Section 6.1. Also, the eleven variations were intended to cover realistic changes in design requirements naval architects may encounter in warship design, most notably changes in weapons fit and performance requirements (see *Van Oers [127]* for a discussion).

Test case (-)	Population size (-)	No. Generations (-)	Total no. of designs (-)	Feasible designs (-)	Infeasible designs (-)	Feasible designs (%)	Reason for test case
1	60	672	40320	401	39919	1.0	Concurrently vary three hull types
2	60	649	38940	34	38906	0.1	Use frigate-type hull no. 2
3	60	881	52860	1124	51736	2.1	Use OPV-type hull
4	60	877	52620	398	52222	0.8	Use frigate-type hull no. 1
5	60	551	33060	479	32581	1.4	Increased separation of redundant systems
6	60	551	33060	226	32834	0.7	Decreased separation of redundant systems
7	60	551	33060	198	32862	0.6	Reduced range (3000 NM at 18 knots)
8	60	551	33060	154	32906	0.5	Increased range (6000 NM at 18 knots)
9	60	393	23580	1407	22173	6.0	Changed payload:AAW-frigate
10	60	551	33060	771	32289	2.3	Changed payload: ASW-frigate
11	60	551	33060	1622	31438	4.9	Changed payload: OPV

Table 6.1 – The eleven test-cases performed with the frigate model

The eleven test cases used a single parametric ship description. The variations were carried out in two ways, either changing the properties of individual systems of objects, or by removing systems of objects.

Section 6.2.2 discusses the parametric ship description, i.e., the systems of objects, their packing sequence and the parameter driven changes they undergo. Section 6.2.3 subsequently discusses the results obtained with the search runs in Table 6.1.

Note that the results of the first test case are discussed extensively. The other results are covered only briefly to investigate whether the packing approach can handle the changes in design requirements and to study the effects these changes have on the total set of configurations.

6.2.2 Parametric ship description

Applying the packing approach means developing a parametric ship description of a frigate-type warship. This Section introduces the systems of objects, the packing sequence and the focussing (constraining) of the parametric ship description that was applied. Note that this Section does not provide a thorough discourse on surface warship design (see *Brown [21]*, for instance) including the elaborate process used to balance requirements and ship design until an affordable compromise is achieved (see *Andrews [5]* and *Van Oers [127]*, for instance).

Instead, this investigation has narrower scope and follows the steps outlined in Section 1.2 to develop the parametric ship description: a functional decomposition, the allocation to systems that are subsequently clustered and the modelling of clusters with the object types from Section 4.4.1.

```
- Perform mission

    - Provide platform
        - Access
        - Carrying platform
        - General support
        - Hotel
        - Mobility
        - Survivability

    - Perform operational functions
        - Maintain situational awareness
        - Exercise command & control
        - Perform Maritime security operations
        - Perform Anti-air warfare
        - Perform Anti-surface warfare
        - Perform Anti-submarine warfare
```

Fig. 6.3 – Functional decomposition used for the frigate application

Approach used to develop the description

- **Functional decomposition**. A functional decomposition was developed following the decomposition proposed by *Klein Woud and Stapersma [69]*, who distinguish between two main functions: provision of a suitable platform and provision of operational functions. Each main function is subsequently decomposed into several subfunctions. The resulting decomposition is shown in Fig. 6.3.

- **Allocation to systems**. The functional decomposition listed in Fig. 6.3 must be allocated to systems. Several Royal Netherlands Navy ships were used to determine system shape, size and weight, i.e., the Patrol Ship, the M-class frigate and the Air Defence and Command frigate. Multiple ships are required because not all systems used in the application are present on a single ship, e.g., only the M-class frigate has a towed array sonar. In addition, a system database developed by *Takken [113]* was used for systems not present on Royal Netherlands Navy ships.

- **Clustering of systems**. The clustering of systems was carried out in a similar manner as discussed in Section 4.4.1. Starting point was that those systems most relevant for early stage design, i.e., with the biggest impact on ship dimensions and performances were identified. Another important consideration was that sufficient variation in ship configuration should be possible, i.e., the positions of large systems should be able to change independently of other systems. Obviously, other applications of the packing approach may see different clusterings.

- **Modelling clusters with systems of objects**. The functional decomposition and subsequent allocation and clustering of systems resulted in a collection of 113 systems with 465 objects of the types discussed in Section 4.4.1 that form the basis for the parametric ship description model.

Systems of objects

The process results in a large collection of systems listed in Table 6.3 and 6.4. Of all systems, systems no. 8,9, 12 and 14 are not included in Table 6.3 and 6.4, as these systems consist purely of logical objects that help focus the ship description. Hence, they are discussed later on in this Section (see the discussion on focussing on page 177). Note, other systems of objects also used logical objects (again see page 177 for some examples).

Furthermore, numerous systems commonly found aboard ships are not included in the parametric ship description, despite the large number of objects already used. Examples of such missing systems are lubrication oil tanks, fresh water tanks and ventilation rooms. For practical application of the ship description presented in this Chapter, these systems should be added.

Ballast tanks are also absent from Table 6.3 and 6.4. The reason is that they are generated during the performance predictions when the required amount and location of ballast is assessed. This procedure was discussed in Section 5.3 (see Fig. 5.8).

Discussing all systems in Table 6.3 and 6.4 would take up too much space. Therefore, only a few systems deemed to be the most interesting are considered in more detail in this Section. Also, Figs. B.1 to B.6 in Appendix B offer an 'exploded' view of the systems of objects used in the frigate application.

- **Envelope**. Three different hull shapes were considered; they can undergo changes in size (length, beam and design draught), as well as shape and fullness of fore-body and aft-body. Fig. 6.4 shows them. The envelope was modelled as a single system with a single object that uses six parameters in total to instigate changes (those listed in Table 4.3). The lower and upper bounds of the six parameters are shown in Table 6.2. Also, the topside part of the envelope is allowed to change following Section 4.7.2 (Fig. 4.81).

- **Propulsion plant with uptakes, shafts and propellers**. One of the features of frigates is their relatively large propulsion plant. For the ship description, a Combined Diesel or Gas-Turbine plant was used, as common on Royal Netherlands Navy Frigates.

  It is modelled as four systems of objects (shown in Fig. 6.5): the propellers; the prime movers (i.e., diesel engines, gas turbines), the gearboxes, the shafts, and the uptakes and the downtakes. The four systems consist predominantly of hard objects (representing propellers, gas turbines, gearboxes and diesel engines), together with connection objects (shafts, uptakes and downtakes) and free-space objects representing the unobstructed areas required to take in air and emit exhaust gases.

  The propulsion plant is sized by predicting the required propulsion power (see Section 5.3) using the assumed design draught (see Section 5.5) at cruise and top speed (18 knots and 30 knots, respectively). The appropriate diesel engines, gas turbines and gearboxes and their size, weight and specific fuel consumption are retrieved from a look-up table using the required power. This information is used to determine the appropriate object sizes.

  Note, that, the propulsion system as modelled in the current parametric ship description (i.e., in Fig. 6.5) is considered to be too simple. It should be made more elaborate and use more thorough sizing rules. For example, additional air intakes are necessary to prevent the current intakes from having to penetrate a water-tight bulkhead (alternatively, one could split them above the damage control deck); a similar issue affects the up takes.

Furthermore, air intakes for gas turbines are equipped with filters to reduce damage due to intake of salty air. However, they were not included in the parametric ship description.

Parameter no.	Changes	Type	Lower bound	Upper bound	Meaning
1	Hull type	Discrete	1	3	1 → OPV-type hull 2 → Frigate-type hull no. 1 3 → Frigate-type hull no. 2
2	Length over all	Continuous	100	140	–
3	Maximum beam	Continuous	12	18	–
4	Design draught	Continuous	12	18	–
5	Stretching of bow	Continuous	0	1	–
6	Stretching of stern	Continuous	0	1	–

Table 6.2 – Parameters used to change envelope shape and size

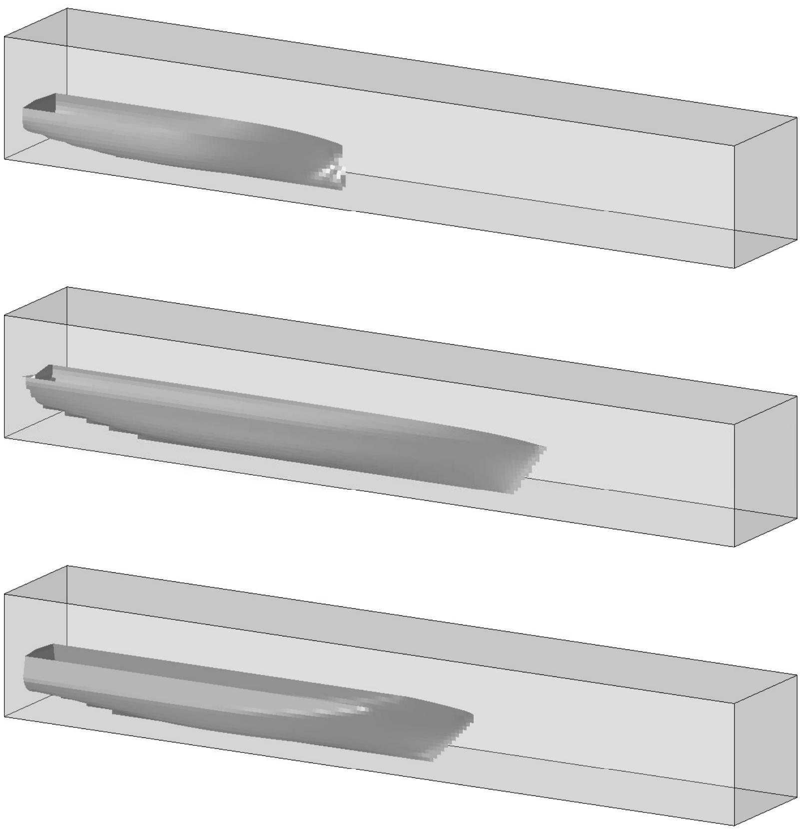


Fig. 6.4 – The three hull shapes used: OPV-type (top), Frigate type no. 1 (middle) and Frigate type no. 2 (bottom)

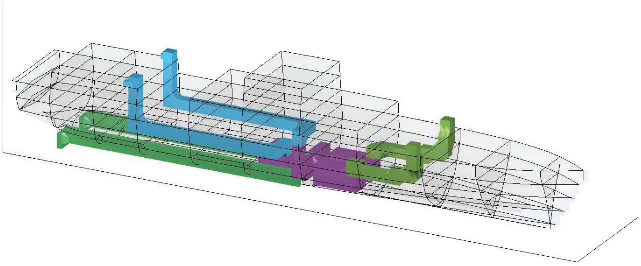


Fig. 6.5 – Propulsion plant defined by four systems consisting of hard objects, connection objects and free space objects (the latter are not shown)

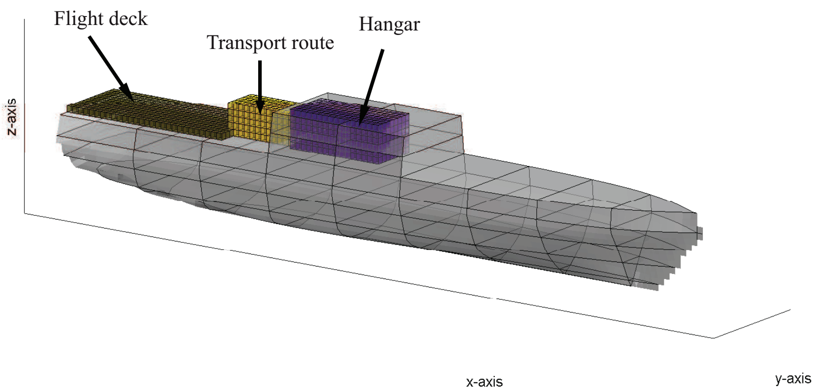


Fig. 6.6 – Flight deck (hard object + free space object) and hangar (hard object) connected by a transport route (connection object)

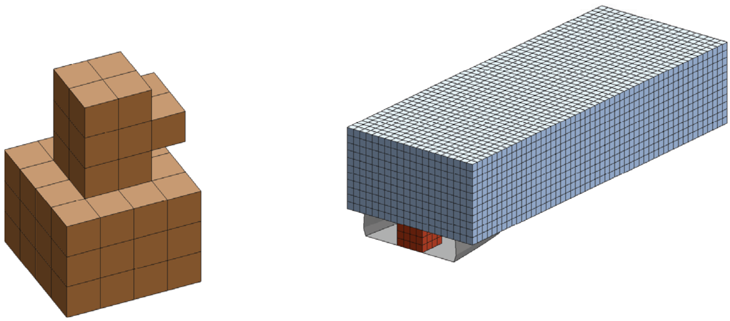


Fig. 6.7 – A gun defined by two hard objects (gun and magazine; left image) and a firing arc (free space object; right image)

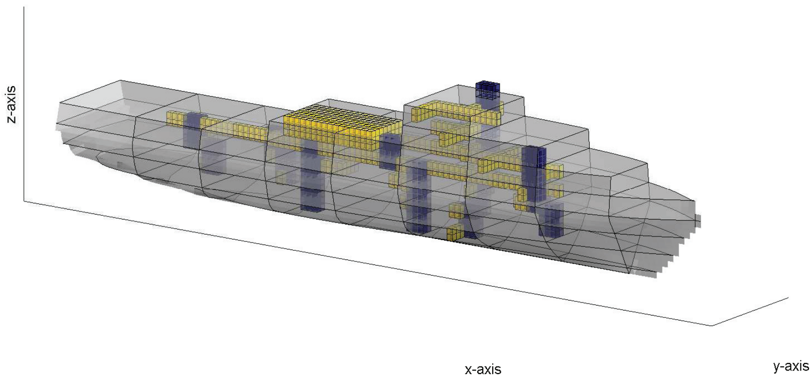


Fig. 6.8 – Three-dimensional network of passageways (yellow) and staircases (blue)

- **Flight deck with hangar**. Another important feature of a frigate are the helicopter deck and hangar used to operate helicopters in a variety of missions. The flight deck and hangar are modelled as two systems of objects (each with their own initial position) to allow the distance between the two to vary.

  The flight deck is modelled by two objects: a hard object representing the flight deck and a free space object representing the unobstructed area required for helicopter operations. The hangar is also modelled by two objects: a hard object represents the hangar, while a connection object represents the transport route used to transport the helicopter to and from the flight deck. Fig. 6.6 shows an example.

- **Three-inch gun**. Warships are equipped with a variety of sensors and weapons. A familiar weapon system found on almost any frigate is the general purpose gun. For the ship description, a 3-inch gun is used. It is modelled as two hard objects (left image in Fig. 6.7), one representing the gun itself, the other systems underneath deck, e.g., the magazine and reloading mechanism. The system also contains a single free space object presenting the required firing arc (180°). The latter extends from the gun's position to the forward boundary of the positioning space, over the full width, as well up towards the upper edge of the positioning space (right image in Fig. 6.7). Two input parameters change the initial x and z-positions. The gun, and the way it is modelled, is representative for many other systems present on warships, e.g., weapons, sensors and even small boats in davits.

- **Passageways & staircases**. The crew requires access to all spaces onboard a warship, which is provided by an elaborate network of passageways and staircases. The network is composed of a large number of connection objects distributed over a similarly large number of systems. The connection objects create or link up to a network of staircases and passageways to ensure all parts of the ship are accessible.

  Combining connection objects with other objects makes it possible to change the position of both objects whilst ensuring the required connectivity. For example, overlap management can alter the position and shape of a soft object (representing a combat information centre) together with a connection object (representing the access route) such that it fits, complies with user-defined constraints, and is connected to the network of passageways and staircases.

The combination of a hard or soft object with a connection to passageways is used frequently, e.g., all accommodation, storage and other internal objects listed in Tables 6.3 and 6.4 are modelled this way. One resulting access network is shown in Fig. 6.8.

Together, the systems of objects in Table 6.3 and 6.4 describe the entire ship, e.g., as in the two examples in Fig. 6.1. Changing the initial positions, shapes and sizes of the systems of objects results allows the search process to generate configurations that comply with the non-negotiable requirements. Section 6.2.3 discusses these results.

Packing sequence

A set of systems of objects is not enough, as one must also establish a packing sequence. The order of packing used for the parametric description in this Section is shown by the system numbers in Table 6.3 and 6.4; it is used for all eleven search runs. It follows the generic sequence proposed in Section 4.5.8 after the elaborate discussion on the impact of sequence in Section 4.5. Note that no distinction based on functionality was used, only spatial features such as shape, size and range of available positions were considered when developing this sequence.

The condensed sequence is listed below.

1. Envelope.
2. Subdivision objects, i.e., decks.
3. Constrained systems with hard objects, e.g., propulsion plant.
4. Subdivision objects, i.e,. bulkheads.
5. Soft objects (volume-based), e.g., fuel tanks.
6. Systems with hard objects and free space objects, e.g., bridge and flight deck
7. Systems with hard objects, e.g., diesel generator set.
8. Systems with soft objects (area-based), e.g., accommodation.

In addition to the condensed sequence outlined above, focussing will influence the sequence of packing of specific pairs of objects. For example, a logical object included in system no. 46, a diesel generator, enforces its separation with the other diesel generator systems (no. 43), requiring the former system of objects to be packed later. Focussing is discussed below.

System number	System name	Main function	Sub-function
1	Envelope	Platform	Carrying platform
2	Decks	Platform	Carrying platform
3	Propulsion plant	Platform	Mobility
4	Propellers	Platform	Mobility
5	Rudders	Platform	Mobility
6	Bulkheads	Platform	Carrying platform
7	Slipway for RHIB	Operational	Maritime security
10	NH-90 flight deck	Operational	Anti-submarine / Anti-surface Maritime security
11	NH-90 hangar	Operational	Anti-submarine / Anti-surface Maritime security
13	Torpedo tubes PS & SB	Operational	Anti-submarine
15	Mk. 41 vertical launch system no.1	Operational	Anti-air
16	Passageway connection to VLS	Platform	Access
17	Bridge	Platform	Mobility
18	3” gun fwd. no. 1	Operational	Anti-air / Anti-surface Maritime security
19	20-foot containers	Operational	Maritime security
20	Downtakes routed	Platform	Mobility
21	Uptakes routed	Platform	Mobility
22	Passageway connection to slipway	Platform	Access
23	Towed array sonar	Operational	Anti-submarine
24	Passageway connection to hangar fwd.	Platform	Access
25	Passageway connection to hangar aft	Platform	Access
26	Passageway connection aft engine room	Platform	Access
27	Passageway connection fwd. engine room	Platform	Access
28	Passageway to hangar	Platform	Access
29	Hull mounted sonar	Operational	Anti-submarine
30	Fuel tank bottom 1	Platform	Mobility
31	Fuel tank bottom 2	Platform	Mobility
32	Fuel tank bottom 3	Platform	Mobility
33	Fuel tank bottom 4	Platform	Mobility
34	CIWS RAM fwd.	Operational	Anti-air
36	Harpoon SSM	Operational	Anti-surface
37	CIWS RAM aft	Operational	Anti-air
38	APAR radar	Operational	Situational awareness
39	RHIB + crane SB & PS	Operational	Maritime security
40	RAS positions	Platform	General support
41	Machine guns aft	Operational	Maritime security
42	Machine guns forward	Operational	Maritime security
43	Generator plant 1	Platform	General support
44	Passageway connection generator 1	Platform	Access
46	Generator plant 2	Platform	General support
47	Passageway connection generator 2	Platform	Access
49	Chilled water plant 1	Platform	General support
50	Passageway connection chilled water 1	Platform	Access
52	Chilled water plant 2	Platform	General support
53	Passageway connection chilled water 2	Platform	Access
54	CIC	Operational	Command & Control
56	Sick bay	Platform	Survivability
58	Ship Control Center	Platform	Mobility
60	Galley	Platform	Hotel
62	Longroom	Platform	Hotel

Table 6.3 – Systems 1 to 62 of the configurational model

System number	System name	Main function	Sub-function
63	Recreational Area no. 2	Platform	Hotel
64	Recreational Area no. 1	Platform	Hotel
65	Backup Ship Control Center	Platform	Mobility
66	Back-up CIC	Operational	Command & Control
67	Damage control centre 1	Platform	Survivability
69	Damage control centre 2	Platform	Survivability
70	Life rafts 1	Platform	Survivability
72	Life rafts 2	Platform	Survivability
73	Accommodation (officers)	Platform	Hotel
74	Accommodation (officers)	Platform	Hotel
75	Accommodation (officers)	Platform	Hotel
76	Accommodation (officers)	Platform	Hotel
77	Accommodation (officers)	Platform	Hotel
78	Accommodation (officers)	Platform	Hotel
79	Accommodation (officers)	Platform	Hotel
80	Accommodation (officers)	Platform	Hotel
81	Accommodation (non-com. officers)	Platform	Hotel
82	Accommodation (non-com. officers)	Platform	Hotel
83	Accommodation (non-com. officers)	Platform	Hotel
84	Accommodation (non-com. officers)	Platform	Hotel
85	Accommodation (non-com. officers)	Platform	Hotel
86	Accommodation (non-com. officers)	Platform	Hotel
87	Accommodation (non-com. officers)	Platform	Hotel
88	Accommodation (non-com. officers)	Platform	Hotel
89	Accommodation (non-com. officers)	Platform	Hotel
90	Accommodation (non-com. officers)	Platform	Hotel
91	Accommodation (ratings)	Platform	Hotel
92	Accommodation (ratings)	Platform	Hotel
93	Accommodation (ratings)	Platform	Hotel
94	Accommodation (ratings)	Platform	Hotel
95	Accommodation (ratings)	Platform	Hotel
96	Accommodation (ratings)	Platform	Hotel
97	Accommodation (ratings)	Platform	Hotel
98	Accommodation (ratings)	Platform	Hotel
99	Accommodation (ratings)	Platform	Hotel
100	Accommodation (ratings)	Platform	Hotel
101	Accommodation (ratings)	Platform	Hotel
102	Accommodation (ratings)	Platform	Hotel
103	Back-up sick bay	Platform	Survivability
104	Fitness	Platform	Hotel
105	Waste disposal	Platform	Hotel
106	Laundry	Platform	Hotel
107	Computing room 1	Operational	Situational awareness
108	Computing room 2	Operational	Situational awareness
109	Storage (food)	Platform	Hotel
110	Storage 1 (general)	Platform	Hotel
111	Storage 2 (general)	Platform	Hotel
112	Storage 3 (general)	Platform	Hotel
113	Topside	Platform	Carrying platform

Table 6.4 – Systems 63 to 113 of the configurational model

Focussing

The five types of focussing from Section 4.6 were all used. One example of each type is discussed below.

- Constrain positions and shapes using sequence. Table 6.3 reveals the flight deck and hangar were packed prior to the uptakes and downtakes of the propulsion plant. This means the fixed sequence may force the latter to be routed around the hangar, if necessary (similar to the situation in the left image in Fig. 4.72).

- Constrain positions using a system of objects. The propulsion plant is modelled by the six hard objects representing diesel engines, gas turbines and gearboxes, in a single system of objects (Fig. 4.23). This ensures that their relative positions remain unchanged when applying design changes for overlap management.

  If required, the relative distance between systems in the propulsion plant could be adjusted using a separate input parameter. This could be used to increase the longitudinal separation of redundant systems, which helps to reduce propulsion plant vulnerability to enemy hits (such separation is used on United States Navy destroyers; see *Friedman [44]*).

- Constrain positions using permissible x, y, z-coordinates. The hangar and flight deck (shown in Fig. 6.6) are both located on decks, i.e., have z-positions that coincide with decks. Moreover, the y-position of the flight deck is fixed at the centre-line position.

- Constrain positions using fixed or bounded initial positions. The initial x-position of the gun system shown in Fig. 6.7 is fixed at the forward boundary of the positioning space. This means it will always be the forward-most weapon system on the ship, following current design practice on Royal Netherlands Navy frigates.

- Constrain relative positions using logical objects. Section 4.6 introduces logical objects, the type of focussing used most extensively in the parametric model. The application of logical objects in the frigate description range from the simple to complex. The former is blocking a compartment housing a waste disposal unit for hygiene reasons to ensure sufficient separation from the sickbay and galley. The latter, uses blockages similar to those shown earlier in Fig. 4.27 to ensure separation of redundant systems. Note that multiple logical objects may be used within a single system, e.g., the flight deck has two logical objects. The first ensures the hangar is located on the same deck, while the second reduces crew exposure to helicopter crashes by blocking accommodation from being located underneath the flight.

  The logical objects in the systems missing in Tables 6.3 and 6.4: 8,9, 12 and 14, had the following purposes. Systems no. 8 and 9 were used as a blockage around the two engine rooms that ensure that no other systems that are sensitive to noise and vibration, e.g., crew accommodation, would be located inside of, or on-top-of, the engine room compartments. System 12 used logical objects in the opposite manner, to constrain the position of the torpedo tubes either directly underneath or on the same deck as the hangar, to facilitate torpedo transfer (both torpedo tubes and the helicopter were assumed to use the same type of torpedo). The logical objects in System no. 14 are similar to those in Systems no. 8 and 9, they prevent other systems, e.g., accommodation, from sharing a compartment with the torpedo tubes, e.g., to reduce the risk for the crew in case of fire.

The application of the five types of focussing helps to steer the parametric ship description towards configurations deemed more ‘promising’ by the naval architect. It should thus help increase the yield of the search process.

Performance requirements & ratings

Sections 5.3 and 5.4 introduced the performance predictions and the ratings derived from them. What is missing are the user-defined performance requirements. They are shown in Table. 6.5.

Performance	Required Value	Unit
Systems of objects that cannot be packed	$= 0$	-
Reserve buoyancy to remain afloat with a damage length of	15	% of length over all
Actual draught - Design draught	≤ 0	m
$\Delta \lVert LCB - LCG \rVert$	≤ 0.5	m
Initial stability (GM) in full and empty load	≥ 1.5	m
Cruise speed	18	knots
Top speed	30	knots
Endurance at cruise speed	4500	nm

Table 6.5 – Performance requirements

The cruise and top speed requirements, as well as the endurance requirements were used to size the propulsion plant objects and fuel tank objects. Moreover, heel was not considered, due to the extensive use of symmetry, and the intended application of packing approach very early on in the design process. This means sufficient time should be available to address unwanted heel when refining configurations (the last step in Fig. 3.6).

The performance requirements from Table 6.5 are combined with the ratings from Section 5.4 to form the single objective and five constraints used by NSGA-II to perform the eleven search runs. The results are discussed in Section 6.2.3.

6.2.3 Results & discussion

Table 6.1 shows the eleven search runs used to test the ability of the packing approach.

All search runs are discussed in this Section. The first search run, however, forms the most elaborate application and offers the best illustration of the ability to change all parts of the configuration. It is therefore discussed quite extensively. The reason is it varied the configuration of a multi-role frigate including discrete (i.e., large) changes in hull type. Its discussion will consider some of the numerical characteristics of the entire set of feasible configurations (such

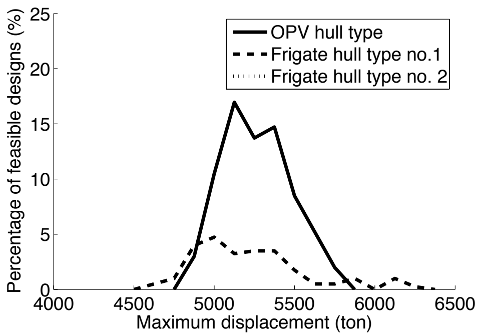


Fig. 6.9 – Distribution of displacement (including ballast)

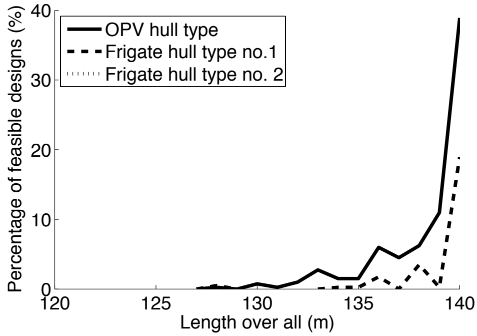


Fig. 6.10 – Distribution of length over all

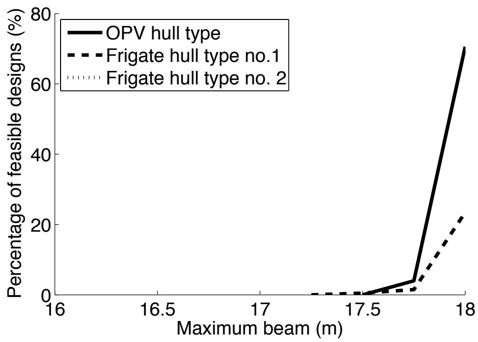


Fig. 6.11 – Distribution of maximum beam

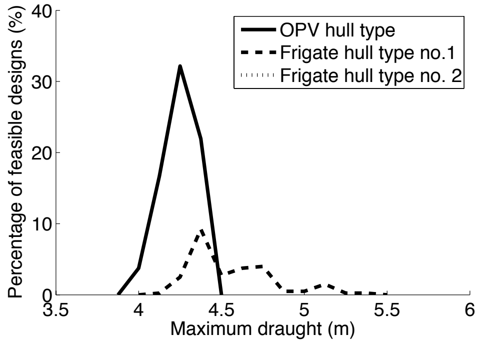


Fig. 6.12 – Distribution of maximum draught

as main dimensions, displacement and initial stability), will compare the arrangements of a small number of configurations and will investigate a single configuration in detail.

The other ten search runs are discussed less extensively. They help illustrate whether the packing approach and search algorithm can investigate changes in design requirements typical of early stage warship design by studying some of the numerical characteristics of the entire set of feasible configurations (such as main dimensions and displacement), instead of treating all their configurational features in detail.

Test case 1: Vary entire configuration to establish ship size

In the first group, the full list of systems shown in Tables 6.3 and 6.4 was packed. Around 40.000 designs were considered, of which around one percent was found to be feasible (see Table 6.1.

The configurations are considered from both a numerical and configurational point of view. The numerical aspects are discussed first and deal with maximum displacement, main dimensions, the use of assumptions regarding draught, and the impact of loading conditions.

- **Numerical aspects: displacement**. Fig. 6.9 shows the maximum displacement (including ballast water) of the feasible configurations. To carry the systems in Tables 6.3 and 6.4, the ships displace between 4500 and 6250 metric tons.

  The large upper bound may seem surprising, since the systems are largely independent of ship size. It is caused by a simple effect: one can always create a bigger ship to carry the same payload, even when this extra space is not necessary. This extra space increases the displacement, since a larger hull and / or superstructure with its own structural weight has to provide it, which explains the largest 6250 tonne displacement. The lower limit is more interesting. It roughly defines the minimum displacement required to carry the systems, fuel and ballast water while complying with constraints on buoyancy, initial stability, trim, speed and endurance.

  There is no guarantee that smaller configurations are impossible, nor is there a guarantee that the smallest configurations do not have flaws that lead to increased displacement as the design is developed further (see Section 6.4.4). Nonetheless, the packing approach gives a size estimate that serves as a starting point for further investigation, while the selection approach from Chapter 7 helps to investigate configurations with lower displacements to see how realistic they are (and whether they suffer from the lack of completeness, Sections 3.5.2 and 3.5.3).

  Considering the impact of hull type reveals several things. No feasible configurations were found that used frigate hull type no. 2; a brief investigation revealed that its much narrower form required increased beam beyond the maximum width of 18 m used in this search run. Also, the majority of configurations have an OPV-type hull though configurations with frigate hull type no. 1 can have a lower displacement (though no investigation was made, this difference in displacement could be caused by a lower resistance at cruise speed for the frigate-type hull no.1, which would result in smaller fuel tanks and -possibly- smaller ballast tanks).

- **Numerical aspects: dimensions**. Fig. 6.10 to 6.12 show the distribution of length, beam and draught of the feasible configurations. The range in dimensions of the feasible designs is considerably smaller than that of the displacement in Fig. 6.9. Length of the feasible designs lies between 127 m and 140 m, with the majority of the ships having a length of 140 m. The beam of the feasible designs is similar and ranges from 17.5 m to 18 m, with almost all ships having a beam of 18 m (though the lower bound for the beam is much lower, see Table 6.2 and Figs. 6.56 and 6.57). Maximum draught is more diverse, varying between 3.9 m and 5.4 m. Draught variation is likely caused by the ballasting calculation, which uses a variable amount of ballast water to ensure that constraints on trim, buoyancy and initial stability from Section 5.4.2 are met, at the cost of increasing the draught of the ship.

  The impact of hull type is again considerable. Configurations with the OPV-type hull have smaller dimensions in general, i.e., length, beam and draught. Though present, the variation in size is fairly limited and Section 6.4.2 therefore investigates this further.

- **Numerical aspects: assumed draught**. Section 5.5 discussed how an assumed design draught is used to predict required propulsion power and reserve buoyancy. The assumption is safeguarded by a constraint (Section 5.4.2). However, it is still worthwhile to check if the assumed draught is not exceeded. Fig. 6.13 shows the design draught (assumed) and the actual draught of the configurations. It shows how the assumed draught is not exceeded and thus that the assumption is indeed safeguarded by the constraint.

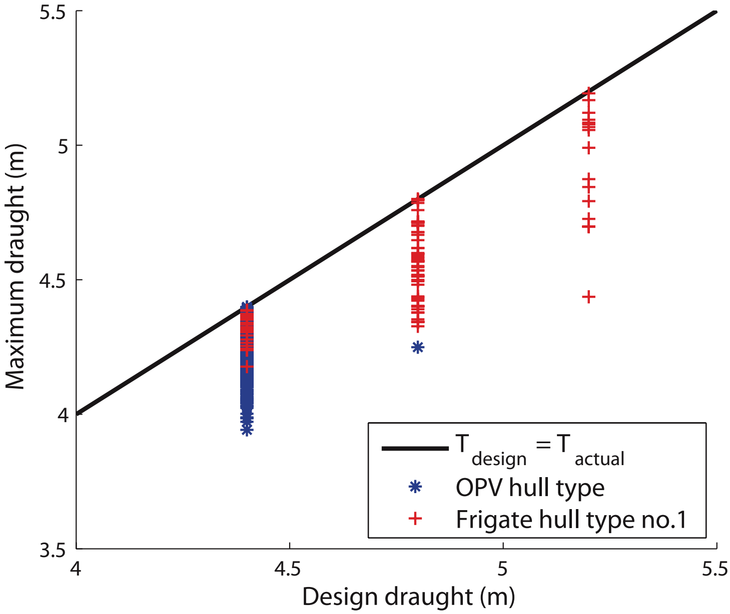


Fig. 6.13 – Design draught vs. actual draught

Note, the vertical lines occur because design draught changed in discrete steps of around 0.5 m.

Fig. 6.13 also shows configurations with excess buoyancy. This extra buoyancy -as well as excess stability and excess space- provides some 'margin for error', and is extremely important to address the lack of completeness raised in Section 3.5.2 and 3.5.3. For example, the reserve buoyancy calculation in Section 4.4.5 (Fig. 4.48) and Section 5.3 does not consider damaged stability. Meeting damage stability requirements can be facilitated by increasing reserve buoyancy (and providing excess stability), which could mean using configurations with a small but sufficient difference between design draught and actual draught (0.25 m, perhaps).

- **Numerical aspects: impact of loading conditions**. Section 5.3 discussed how hydrostatic calculations may use a ballasting calculation to meet constraints on buoyancy, trim and stability in different loading conditions. Two loading conditions are considered: fully loaded (100 % weapons and 100 % fuel), and empty fuel condition (loaded with 100 % weapons and 10% fuel). Fig. 6.14 visualises displacement -including ballast water- in both conditions.

  Fig. 6.14 reveals nearly all configurations have a larger displacement in full load condition than in empty fuel condition. Some large configurations with a frigate-shaped hull constitute an exception, though (a brief investigation was conducted as to why, but no clear cause could be established). This result means that most configurations adhere to a sound naval architectural principle: do not have a larger displacement in empty condition than in full load condition.

  Note, the limits of the ballasting approach, including why it might result in designs that need ballast water in full load condition, were discussed in Section 5.3.

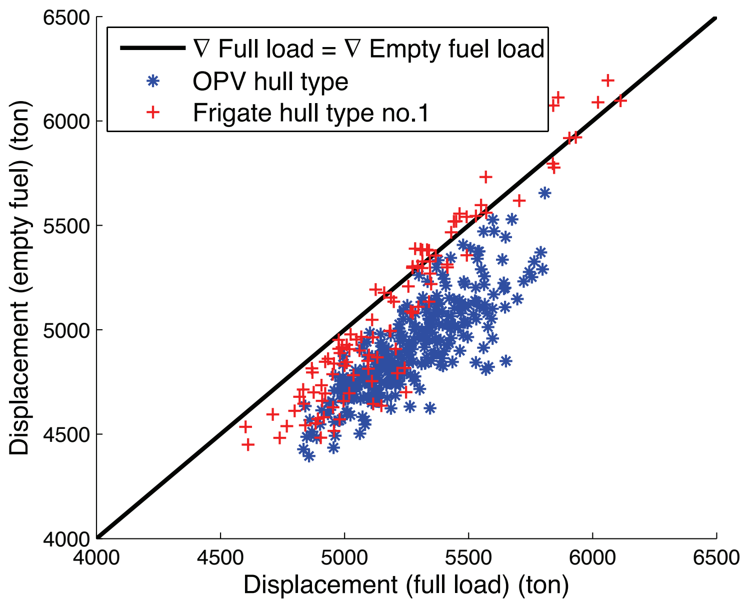


Fig. 6.14 – Displacement in both loading conditions

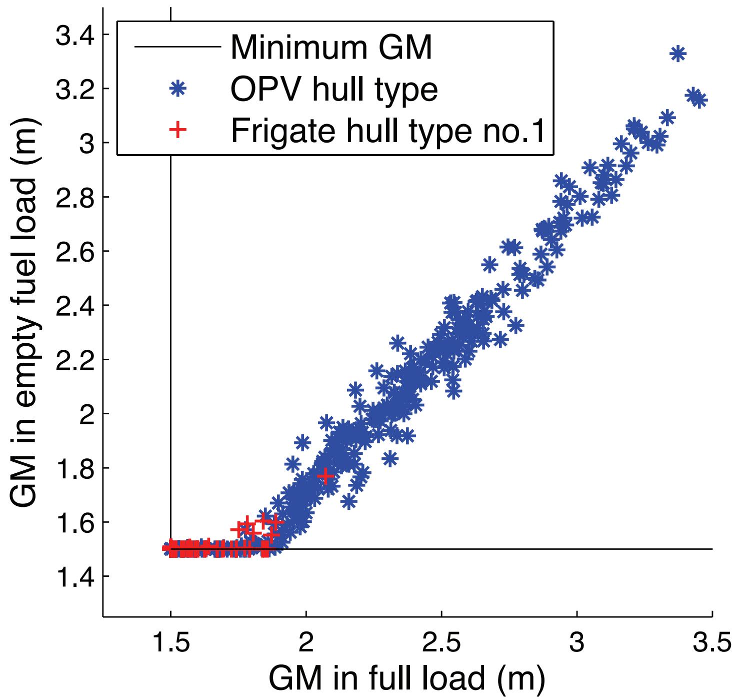


Fig. 6.15 – Initial stability (GM) in both loading conditions

- **Numerical aspects: initial stability**. Initial stability is another performance with a non-negotiable requirement. Fig. 6.15 shows the meta-centric height (GM) for the two loading conditions discussed above. All configurations, in all loading conditions, comply with the minimum GM of 1.5 m (Section 5.4.2).

  Two other things are also of interest. First, GM in empty condition is typically lower than that in full load condition. Second, configurations with a GM equal to 1.5 m typically require ballast water to comply with the stability requirements. Again, the use of ballast water is needed more frequently in empty condition. This is shown by the many designs located on the horizontal line at $GM_{empty} = 1.5\ m$ in Fig. 6.15.

- **Comparing configurations**. Next investigation of the configurations is a simple comparison of the overall arrangement of eight configurations shown in Fig. 6.16 to 6.25. The colours indicate function, following *Andrews et al. [10]*: fight (red), move (yellow), hotel (green) and access (pink for passageways and purple for staircases). Refer to Appendix B for an overview of the systems.

  The comparison of Fig. 6.16 to 6.25 reveals numerous important features. First, the configurations differ in both main dimensions and hull type, e.g., compare Fig. 6.17 to Fig. 6.20. As discussed previously, the variation in size is limited (length is most apparent in Fig. 6.16 to 6.25) and will be discussed further in Section 6.4. Second, the superstructure shape and size also varies considerably between large, e.g., Fig. 6.18 and minimal, e.g., Fig. 6.24; several configurations also have a twin-island superstructure. Deck height of the tank top and bulkhead spacing also vary, e.g., compare Fig. 6.19 and Fig. 6.24. Third, the positions of systems also vary considerably. For example, the flight deck moves both longitudinally and vertically, as does the radar mast. One configuration, e.g., 6.22, has the exhausts routed towards the stern, instead of towards the more traditional position amidships. The internal arrangement also shows marked differences, e.g,. witness the changes in fight-related (red) and hotel-related (green) systems inside the ship. Fourth, in general, the overall configuration include those typical for frigates, e.g., Fig. 6.16, 6.19 and 6.21, as well as covers more unconventional configurations as in Fig. 6.24 and 6.23 (though twin-island frigates were discussed both in *Leopold and Reuter [73]* and in *Brown [21]*).

  In summary, Fig. 6.16 to 6.25 illustrate the packing approach is indeed able to change all parts of the ship configuration, which was one of the issues raised in Section 6.1.

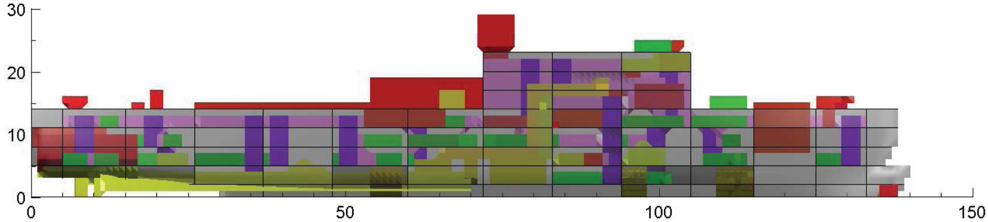


Fig. 6.16 – Design no. 9897

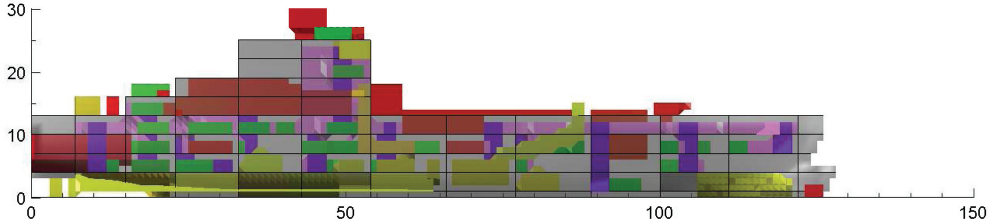


Fig. 6.17 – Design no. 9805

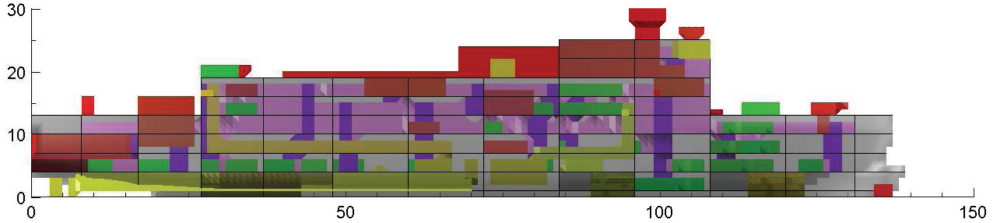


Fig. 6.18 – Design no. 24841

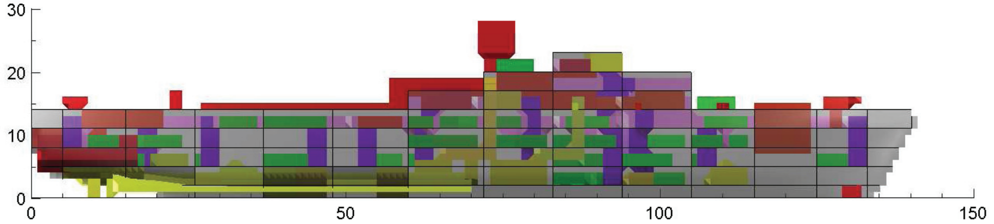


Fig. 6.19 – Design no. 12964

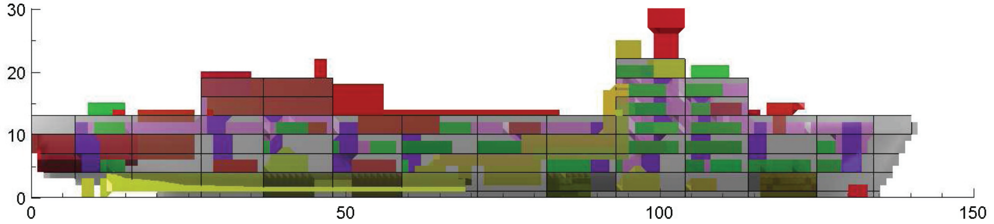


Fig. 6.20 – Design no. 15732

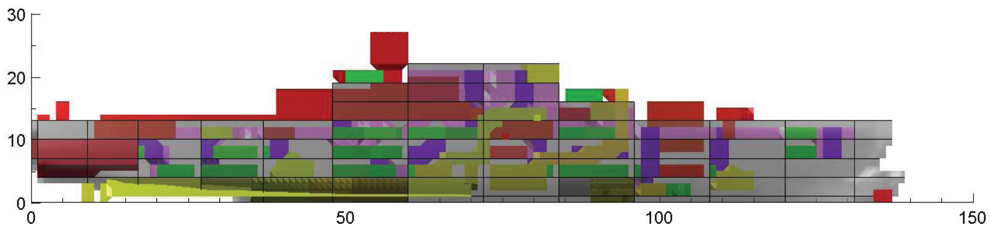


Fig. 6.21 – Design no. 16748

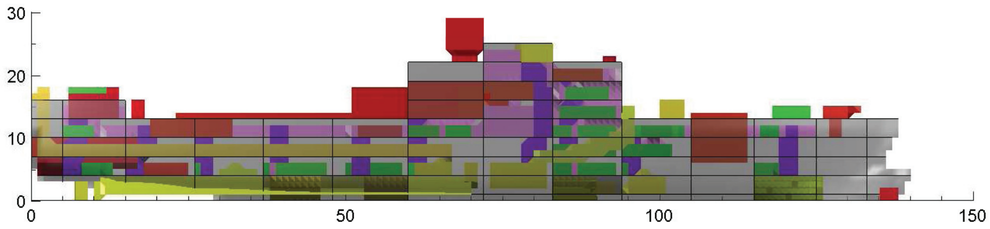


Fig. 6.22 – Design no. 22479

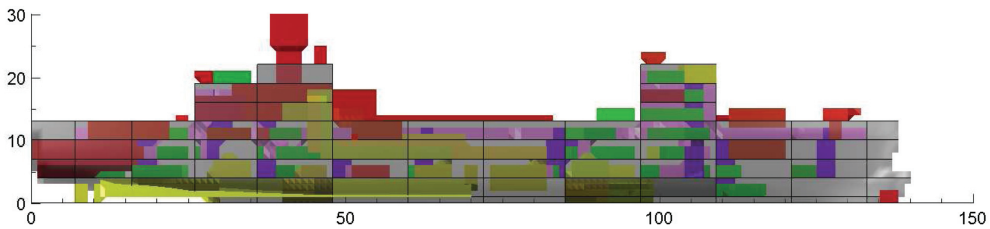


Fig. 6.23 – Design no. 29961

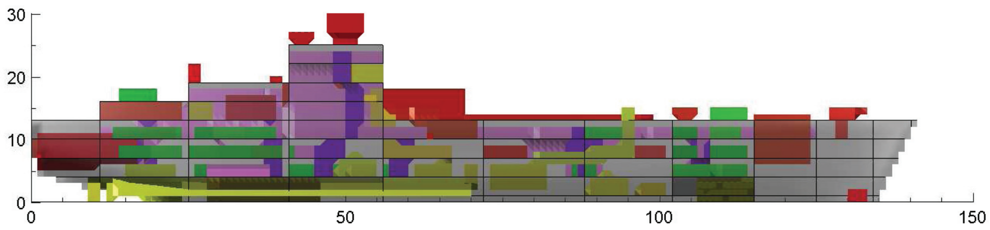


Fig. 6.24 – Design no. 34106

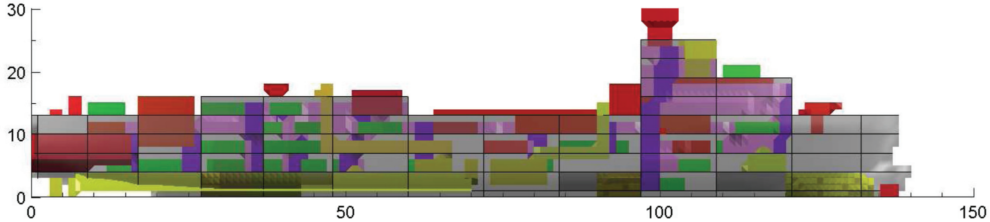


Fig. 6.25 – Design no. 11478

- **Detailed investigation of a single configuration**. The configurations shown in Fig. 6.16 to 6.25 helped to illustrate how the packing approach can indeed change all parts of the ship configuration. Still, further investigation of a single design helps establish more insight in the results. The design: no. 23038, was selected by the author using the selection approach to be introduced in Chapter 7. It is shown in Fig. 6.26.

  Note, the particular design in Fig. 6.26 is not a so-called 'base-line' design which the parametric ship description uses as the basis for variations. The reason is that the parametric ship description discussed in Chapter 4 avoids the use of 'base-line' designs to be able to make large and concurrent changes to the entire ship description (unlike *Burger and Horner [23]*, for instance). As such, the design in Fig. 6.26 is merely one out of the 401 feasible designs that were generated during the search process.

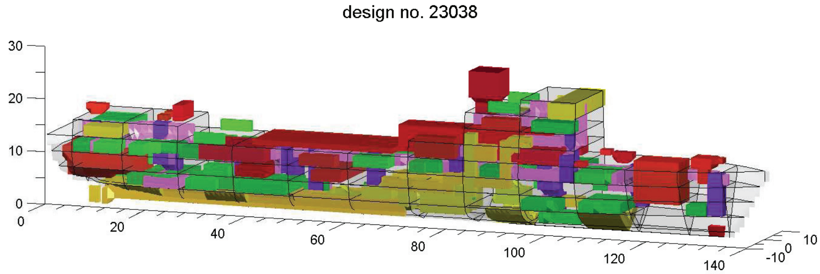
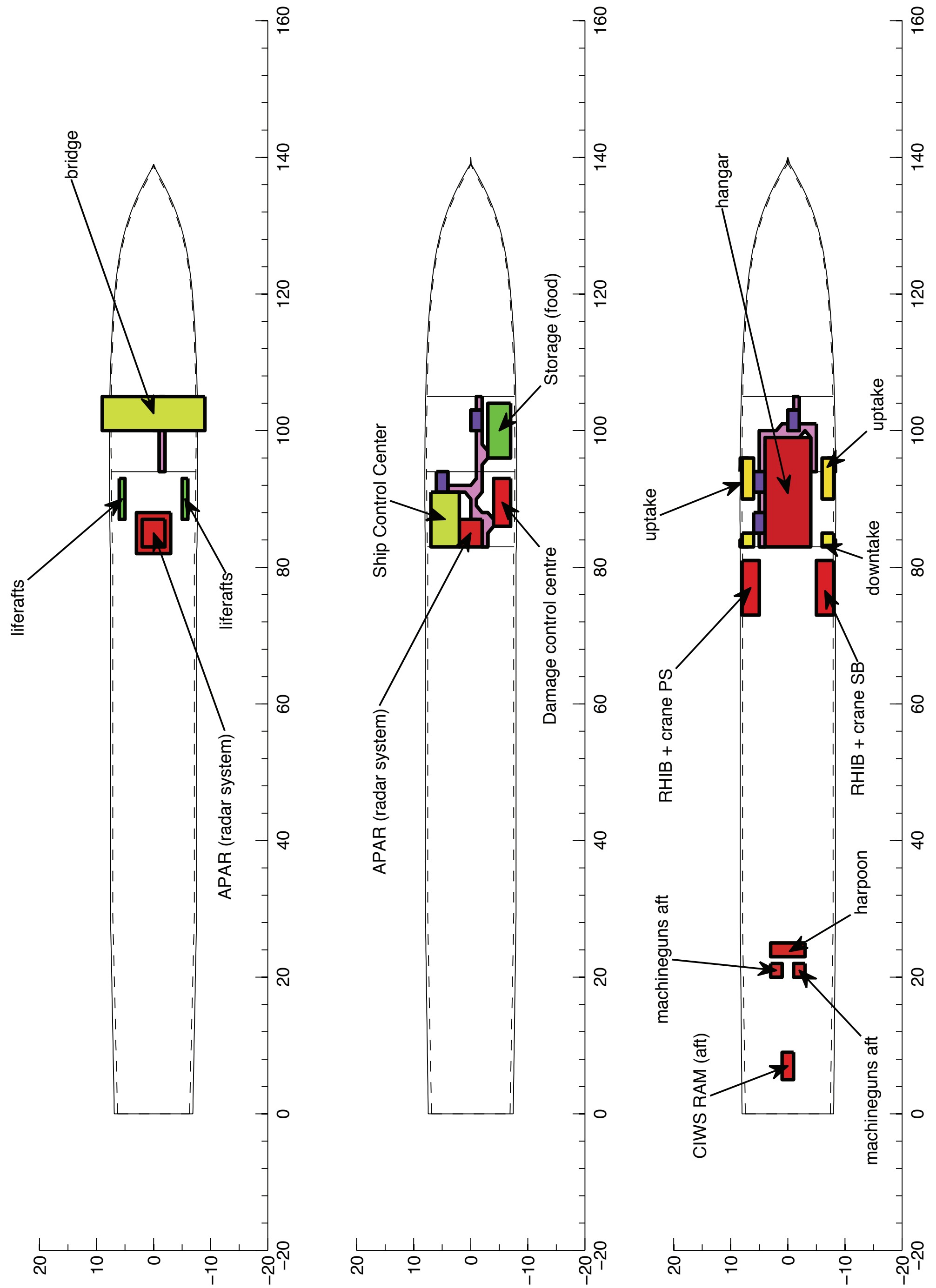


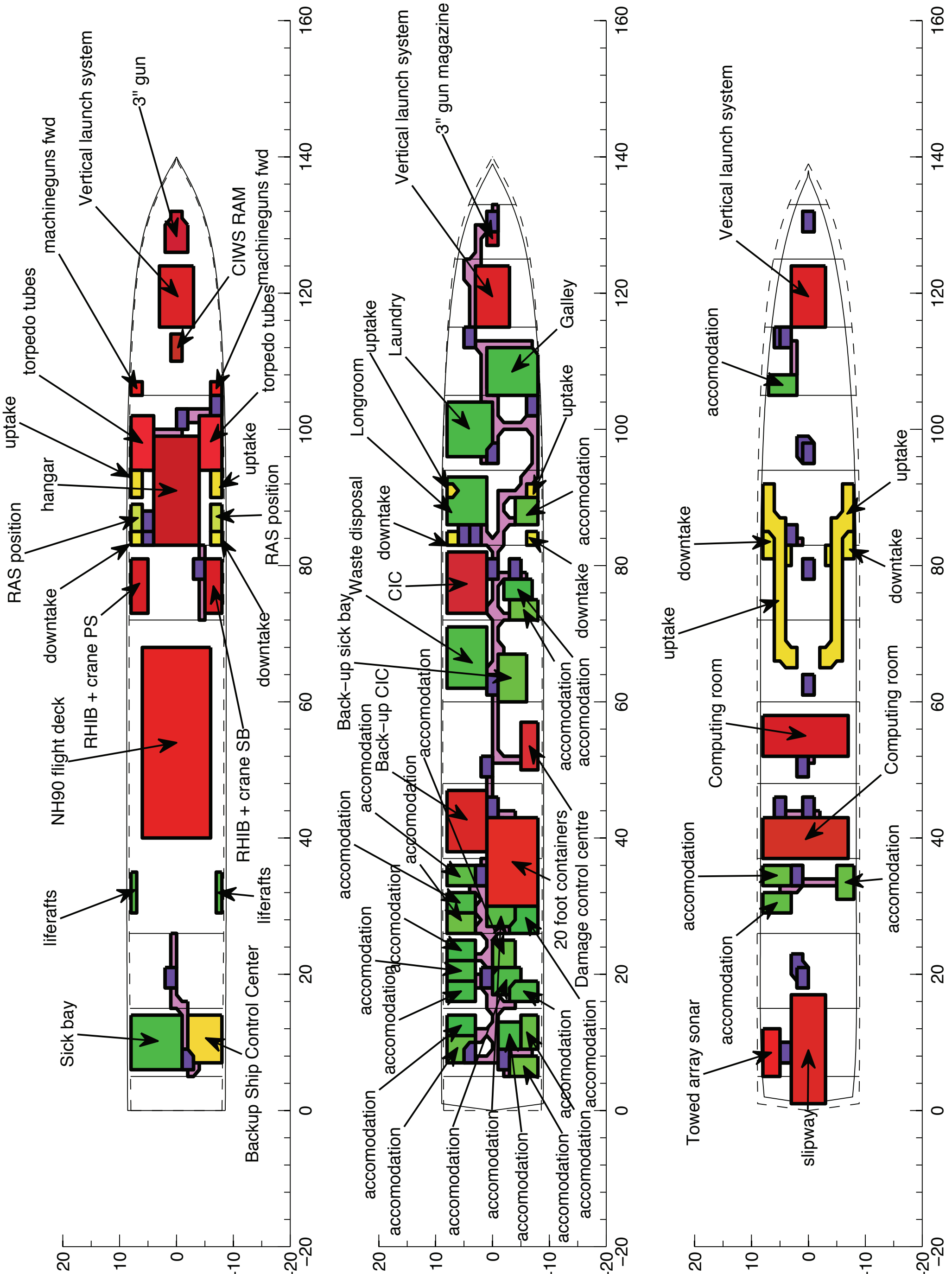
Fig. 6.26 – Design: no. 23038

The configuration in Fig. 6.26 has a reasonably conventional layout for a frigate. Bridge, forward close-in weapon system, VLS and gun are all placed near the bow. The flight deck is moved slightly towards amidships. Still, the raised deck at the stern, which harbours the sickbay on the inside and has the aft close-in weapon system and Harpoon anti-ship-missiles located on top, is to be avoided. The forward superstructure houses the hangar underneath the bridge.

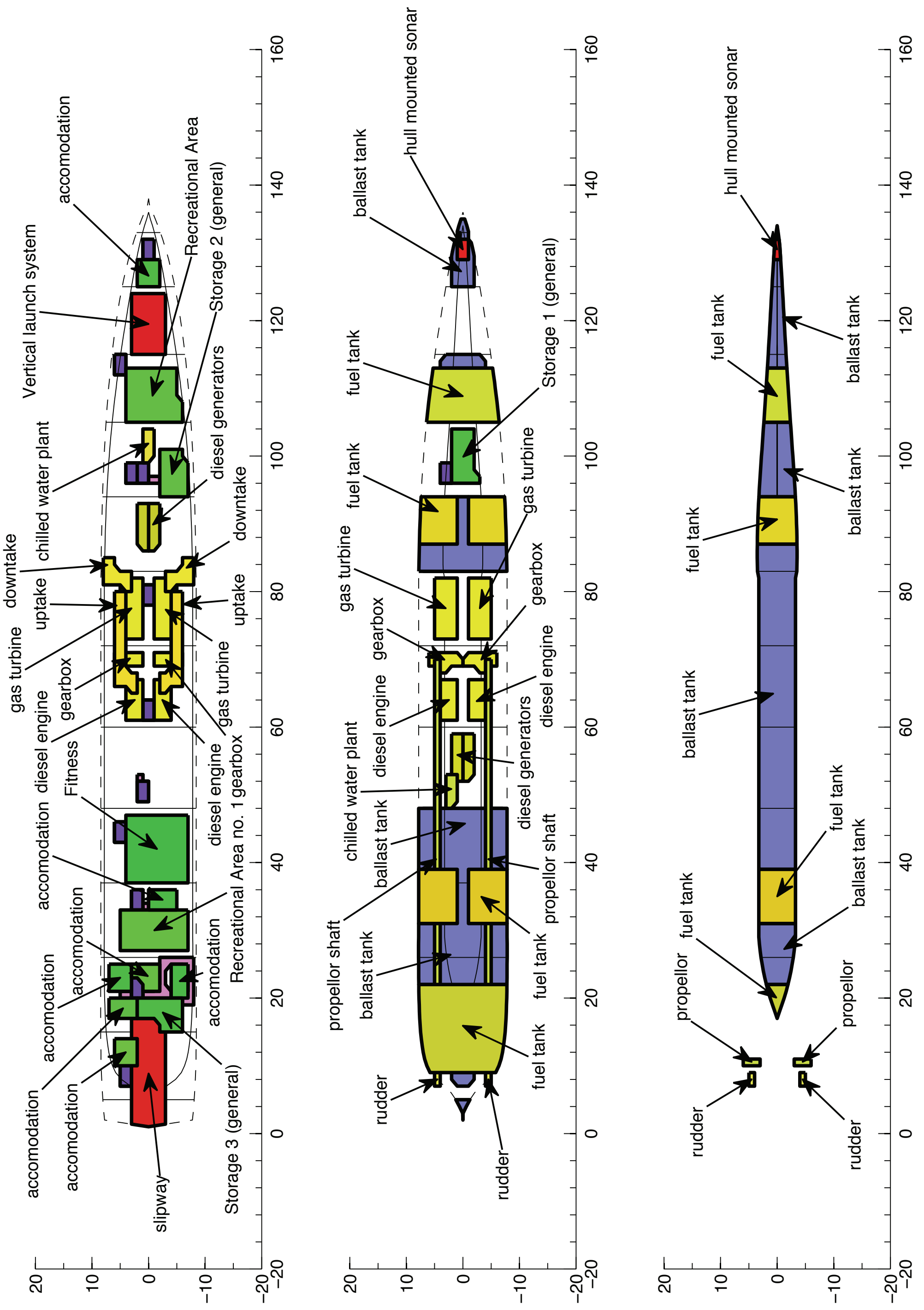
The interior arrangement is also important and shown in Figs. 6.27 to 6.29. For instance, Fig. 6.28 reveals that the torpedo rooms are located adjacent to the hangar and that the RHIBs[1] and accompanying cranes are located directly forward of the flight deck near amidships (which improves their operability in high sea-states). The deck below the

[1]RHIB stands for Rigid Hull Inflatable Boat


bridge
liferafts
liferafts
APAR (radar system)
Ship Control Center
Storage (food)
APAR (radar system)
Damage control centre
hangar
uptake
uptake
downtake
RHIB + crane PS
RHIB + crane SB
harpoon
machineguns aft
machineguns aft
CIWS RAM (aft)
−20
0
20
40
60
80
100
120
140
160
20
10
0
−10
−20



machineguns fwd
Vertical launch system
3" gun
torpedo tubes
uptake
hangar
RAS position
downtake
RHIB + crane PS
NH90 flight deck
liferafts
Sick bay
Backup Ship Control Center
RHIB + crane SB
CIWS RAM
machineguns fwd
torpedo tubes
uptake
RAS position
downtake
liferafts
Vertical launch system
3" gun magazine
Longroom
uptake
Laundry
Galley
Waste disposal
downtake
CIC
Back-up sick bay
Back-up CIC
accomodation
20 foot containers
Damage control centre
Towed array sonar
slipway
Computing room


accomodation
Vertical launch system
Recreational Area
Storage 2 (general)
chilled water plant
diesel generators
downtake
uptake
gas turbine
gearbox
diesel engine
Fitness
accomodation
accomodation
accomodation
accomodation
accomodation
Recreational Area no. 1
slipway
Storage 3 (general)
ballast tank
hull mounted sonar
fuel tank
Storage 1 (general)
gas turbine
gearbox
diesel engine
chilled water plant
diesel generators
propellor shaft
ballast tank
fuel tank
rudder
propellor
160
140
120
100
80
60
40
20
0
−20
20
10
0
−10
−20

hangar houses the 20-foot containers, the CIC and the main hotel facilities, e.g., galley, longroom, laundry and waste disposal. This deck also houses crew cabins near the stern.

Fig. 6.27 to 6.29 also reveal issues in need of improvement. The accommodation near the bow of the ship in Fig. 6.29 will suffer from considerable accelerations in a seaway. Similarly, the location of the back-up sickbay opposite the waste disposal area might also be reconsidered during a later stage in the design process (see the discussion on the need for refinement in Section 3.4.2).

One last note, the network of passageways (in pink) and staircases (in purple) provide humans access to the entire ship. This is even the case of the apparent obstruction provided by twenty-foot containers, the passageway runs in-between the containers, however, this was not shown by the visualisation process used to generate Figs. 6.27 to 6.29.

The discussion of Test case 1 illustrates the packing approach is able to generate a large and diverse set of feasible configurations. The configurations differ with respect to their main dimensions, shape of hull and superstructure, their position of decks and bulkheads and, crucially, the positions of objects. Some of the ship configurations may be considered traditional and look similar to existing frigates, while others look more unconventional. Furthermore, a detailed investigation discussed the internal arrangement of a single design and highlighted both desirable and less desirable features.

Most importantly, the packing approach is able to determine the size of the frigate required to fit and carry all systems listed in Table 6.3 and 6.4, while adhering to the requirements in Table 6.5 as well as a set of user-defined constraints (discussed in Section 5.4.2). The accuracy of the size estimate is discussed in Section 6.4.4.

The results from the other test cases are discussed below. They should help establish the ability of the parametric ship description and search algorithm to investigate the impact of changes in design requirements.

Test case 2 to 11: investigating changes in requirements

The discussion of the ten other test cases looks into the ability of the packing approach and search algorithm to investigate the impact of changes in design requirements typical for early stage warship design. The ten test cases are divided into four groups: variation of hull type, variation of the degree of separation of redundant systems, variation in range and variation in weapons and sensor suite.

The four groups are discussed below, with a focus on the numerical characteristics of the set of feasible designs. The reason is to investigate whether the impact on overall ship size can be established and, whether it can be explained. The investigation does not discuss configurational characteristics of the designs, as such features of designs were covered already in the discussion of Test case 1.

- **Impact of hull type (Test case 2 to 4)**. Test case 1 varied hull type during the search process. Test cases two to four repeat the hull type variation, but only use a single hull type per search run, to get a better understanding on its impact on ship size and fuel consumption. Fig. 6.30 shows the range of displacement for the three search runs. Fig. 6.31 shows the fuel capacity required to sail 4500 nautical miles at 18 knots.

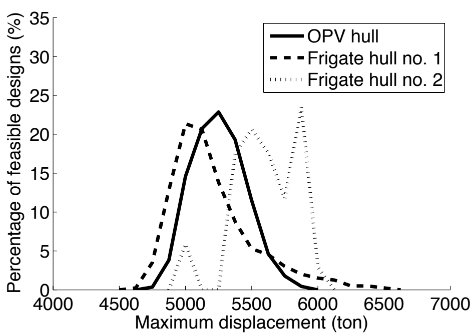


Fig. 6.30 – Distribution of displacement

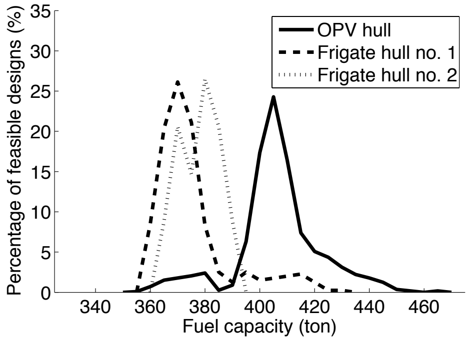


Fig. 6.31 – Distribution of fuel capacity

Displacement for all three hull types lies between 4700 metric tonne and 6500 metric tonne (Fig. 6.30). However, the distribution of displacement differs considerably. Frigate-type hull no. 1 has the most designs with smaller displacement (but also some configurations with a very large displacement). The OPV-type hull has a more narrow distribution of displacements, but is, on average, a few hundred tonne heavier than the frigate-type hull no. 1. Frigate-type hull no. 2 has, on average, the largest displacement of all, but it is thought to be related to the small number of feasible configurations found (see Table 6.1).

The amount of fuel required to sail 4500 nautical miles at 18 knots in Fig. 6.31 also differs markedly. It shows that all hull types meet this design requirement with a fuel load of around 360 tons of fuel. However, frigate type hull no. 1 does have, on average a lower resistance (i.e., requires 380 tons of fuel or less), while the OPV-type hull form requires 400 to 420 tons of fuel. Frigate type hull no. 2 lies in-between the two other hull forms.

Fig. 6.30 and Fig. 6.31 can be used to reduce the different hull type options the naval architects face (Section 1.3). Hence, for the research in this dissertation, frigate-type hull no. 1 is selected as the hull type to use in the remaining test cases: five to eleven, for reasons of a lower resistance at cruise speed and a correspondingly smaller displacement.

- **Impact of separation of redundant systems (Test case 4 to 6)**. Warships are subject to a wide range of enemy threats and the frigate-type warships studied in this Section may need to survive a hit and continue with their mission. Though vulnerability reduction is a wide-ranging topic, one simple measure is the duplication and separation of redundant systems to ensure that, if one system is hit, the other system survives.

  The required separation degree of redundant systems is driven by the size of the damage caused by a hit (defined here by the number of compartments destroyed). A hit by a weapon with a smaller warhead results in a smaller damage size, and thus might allow a reduction of the required degree of separation. Trying to withstand it could thus be possible (potentially) with a smaller ship size. A hit by a weapon with a larger warhead causes more damage. Withstanding such a larger weapon may therefore necessitate an increase in ship size.

System number	Ensures separation between
35	Forward and aft CIWS
45	Generator sets
48	Chilled water plants
51	Chilled water plants
55	Combat Information Centres
57	Ship control centres
59	Damage control centres
71	Life rafts

Table 6.6 – List of redundant systems separated for vulnerability reduction

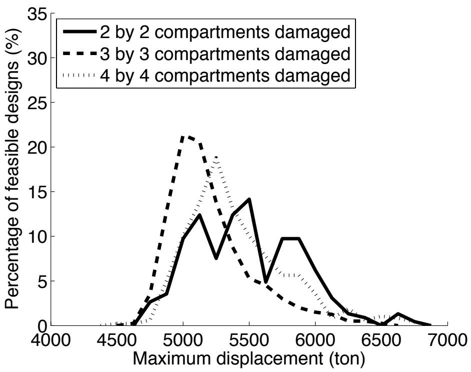


Fig. 6.32 – Distribution of displacement

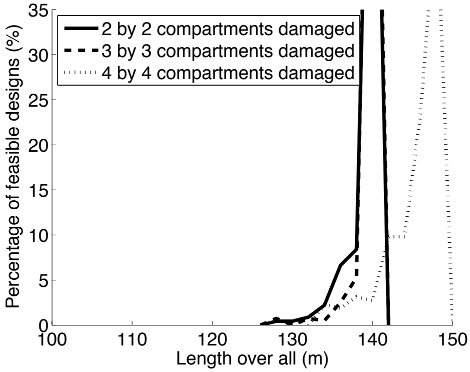


Fig. 6.33 – Distribution of length

Note, this test case does not presume that a navy can choose the enemy weapons that will hit the ship. Instead, the variation in damage size is merely conducted as an exercise to study how the need to survive hits by particular weapons might impact the ship's design.

The parametric ship description uses logical objects to separate the redundant systems listed in Table 6.6. Size of the separation varied between 2-by-2, 3-by-3 and 4-by-4 compartments damaged, to study the impact on overall ship size.

Several assumptions were used. First, damage was assumed to extend over the full width of the ship. Second, uptakes and downtakes of diesel generators were not defined and their considerable impact on the configuration neglected. Third, different damage sizes only affected separation, i.e, they were not used to adjust the damage length considered in the reserve buoyancy prediction, nor was the impact on the routing of the cables, piping and ducts of distribution networks considered (e.g., by using the vulnerability assessment method for distribution networks presented in *Van Ingen [125]*). Fourth, the separation of systems of the propulsion plant was not varied. These assumptions should be addressed in more detailed studies to draw more firm conclusions on the impact of damage size on overall ship size.

Assuming damage of the full ship width means separation affects only the longitudinal and vertical position of objects. The impact of damage size will be established on both length (due to the longitudinal separation) as well as displacement. These are shown in Fig. 6.32 and 6.33, respectively.

Intriguingly, the smallest configuration for the 2-by-2 damage size has a displacement around 4600 tonne, while the smallest configurations for 3-by-3 and 4-by-4 damage sizes weigh 4500 and 4300 tons respectively. Generally though, the impact of the three damage sizes on the distribution of ship displacement is considered negligible (taking into account the assumptions listed above); the displacement distribution is broadly similar and each set configurations displacing in-between 4250 and 7000 tons.

The impact on length is similarly small. Most obvious difference in length is the higher user-defined upper-bound for the largest damage size. This increase from 140 to 150 m enables the ship to grow -if necessary- to cope with the extra separation. Still, Fig. 6.33 shows this extra length is not necessary, as the minimum length for all damage sizes lies below the 140 m. The minimum length for each damage size does differ, the two smallest damages result in a length of 126 m, while the 4-by-4 damage has a minimum length that is slightly higher: 132 m.

The limited impact of separation of redundant systems can be explained using Fig. 4.27. The required separation is small compared to overall ship size and, hence, ship size is not determined by the need to separate redundant systems (Fig. 6.32 and 6.33). Instead, the size is determined by the length needed to accommodate all sensors and weapons residing topside (*Friedman [44]* and *Andrews [4]*).

- **Impact of range at cruise speed (Test case 4, 7 and 8)**. Warships have to cover large distances to reach the operating area. Depending on whether they operate alone or in a task group, they may or may not have support in the form of replenishment vessels that can refuel warships at sea. The need for replenishment can by adjusted by increasing or decreasing range at cruise speed. Three ranges were considered: 3000, 4500 and 6000 nautical miles at a cruise speed of 18 knots.

  Fig. 6.34 illustrates the impact of the fuel capacity of the ship. This obviously affects the maximum displacement of the ship, which is shown in Fig. 6.35. The distribution of displacement in Fig. 6.35 shows the configurations with the smallest range have the smallest displacement. The opposite is not generally true; all range requirements can be met by configurations over 6000 tons in displacement (again, one can always make the ship bigger).

  Altering range at cruise speed has knock-on effects, e.g., it may affect fuel efficiency and required ballast tank capacity. Fig. 6.36 illustrates this. The amount of fuel required

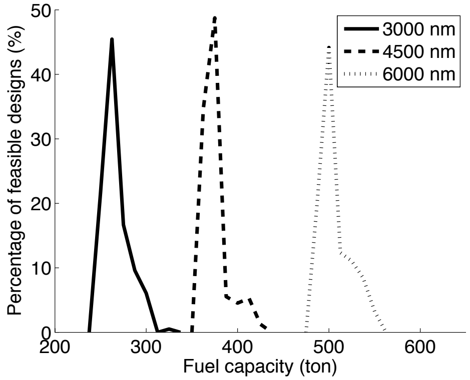


Fig. 6.34 – Distribution of fuel capacity

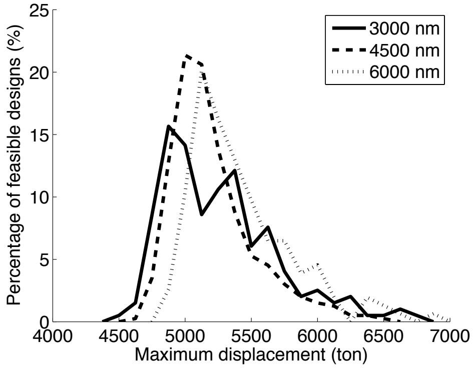


Fig. 6.35 – Distribution of displacement

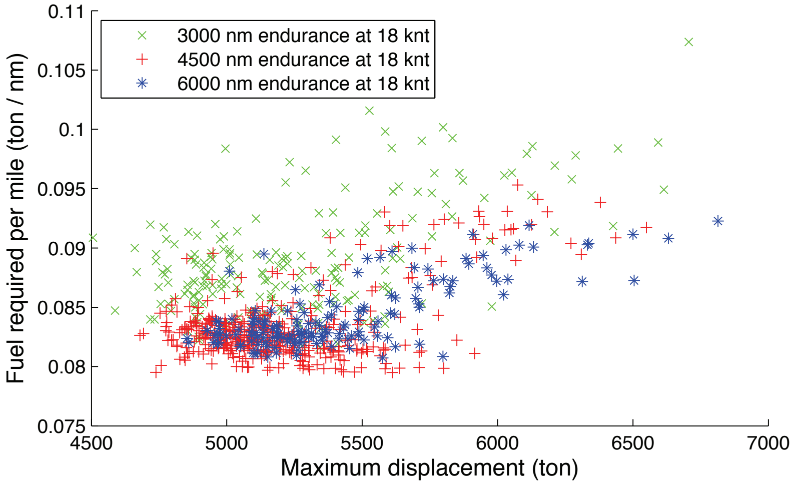


Fig. 6.36 – Fuel per mile plotted against displacement (different ranges)

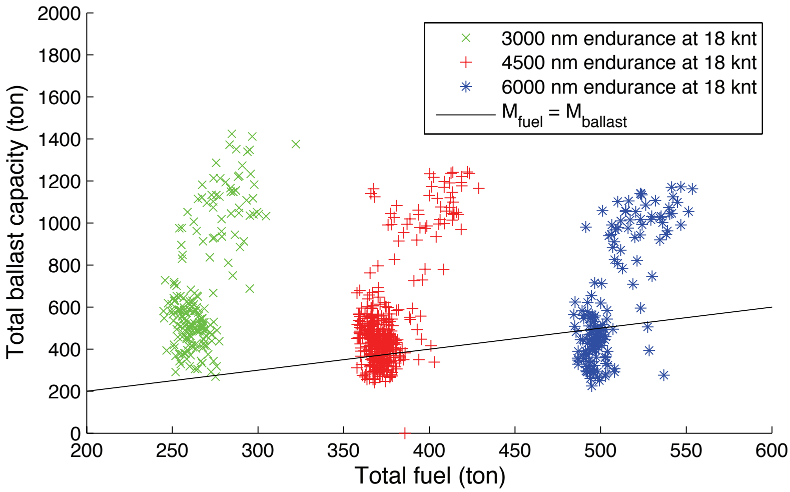


Fig. 6.37 – Ballast capacity plotted against fuel capacity

to sail 1 nautical mile used by configurations with 4500 nautical miles and 6000 nautical miles endurance overlap, while configurations with a 3000 nautical miles endurance require more fuel per mile.

Fig. 6.37 shows the probable cause. Configurations with a 3000 nautical miles range use additional ballast water to compensate for the reduced amount of fuel. Hence, they are found above the line indicating equal ballast and fuel capacity. Since the smaller fuel tanks are located low in the ship, the extra water ballast is needed to comply with the constraint on initial stability in Section 5.4.2. The test cases with 4500 nautical miles and 6000 nautical miles range requirements have to use less ballast water and have numerous configurations below line of equivalent fuel tank and ballast tank capacity.

- **Impact of payload changes (Test case 4, 9, 10 and 11)**. The last three test cases looked into payload variations. One of the purposes of early stage warship design is balancing of design requirements ('the ambition') against cost ('affordability'). Varying the systems the ship will carry, e.g., her weapons and sensors, offers a good way to do this (as does varying performance requirements, which were investigated previously in this Chapter). Therefore, the packing approach must be able to cope with changes in payload in order to be able to support early stage warship design.

  Payload changes are created by removing or altering particular systems of objects from the complete set of systems that together describe the multi-mission frigate (see Tables 6.3 and 6.4). Three variants were made: an air-defence frigate (AAW), an anti-submarine (ASW) frigate and a large ocean-going patrol vessel (OPV). Table 6.7 lists the changed systems.

  The packing approach uses a simple list with systems of objects (e.g., as in Tables 6.3 and 6.4). This lists facilitates the removal and alteration of systems needed to achieve the payload variations in Table 6.7, without altering the overall packing sequence, overlap rules and overlap management outlined earlier in Chapter 4. As a result, the time required to implement all changes listed in Table 6.7 was around 3 hours in total (mostly spent on altering systems), though the subsequent search runs took several days.

AAW-frigate Systems removed	Name
7	Slipway for RHIB
10	NH-90 flight deck
11	NH-90 hangar
13	Torpedo tubes PS & SB
23	Towed array sonar
29	hull mounted sonar
19	20-foot containers
39	RHIB + crane PS & SB

ASW-frigate Systems removed	Name
7	Slipway for RHIB
34	CIWS RAM fwd.
36	Harpoon SSM
39	RHIB + crane PS & SB
19	20-foot containers

ASW-frigate Systems altered	Name
15	8-cell VLS (replaces 40-cell VLS)
38	Smaller radar (Replaces APAR)

OPV Systems removed	Name
13	Torpedo tubes PS & SB
15	mk. 41 vertical launch system no.1
34, 37	CIWS RAM fwd. & aft
36	Harpoon SSM
23	Towed array sonar
29	Hull mounted sonar
66	Back-up CIC
65	Backup Ship Control Centre
63	Recreational Area no. 2
79, 80	accommodation (officers) 2x
89, 90	accommodation (non-com. officers) 2x
100, 101, 102	accommodation (ratings) 3x
103	Back-up sick bay
112	Storage 3 (general)

OPV Systems altered	Name
38	Smaller radar (Replaces APAR)

Table 6.7 – List of systems removed or altered for payload variations

The three resulting variants differ considerably. First thing to notice from Table 6.1 is the yield (the percentage of feasible designs found during the search process, see Section 3.4.2), which is much higher for the three variants than for the original multi-mission frigate from Test case 4 (the cause: reducing the number of systems in the description makes it easier for the search algorithm to find feasible configurations, will be discussed in Section 6.4.2).

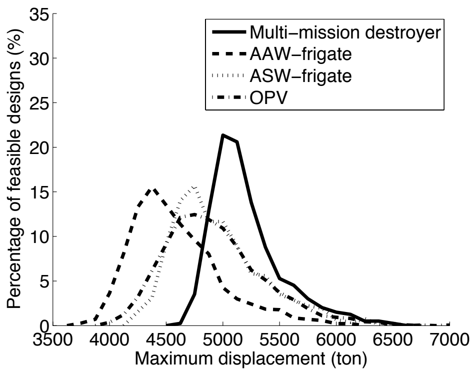


Fig. 6.38 – Distribution of displacement

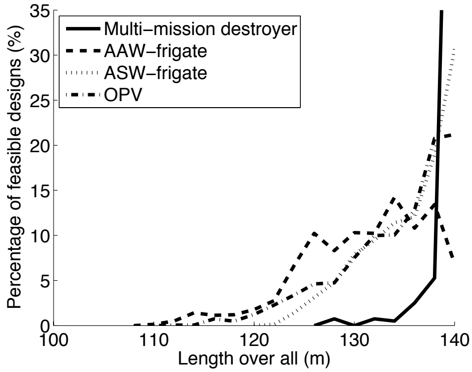


Fig. 6.39 – Distribution of length

Furthermore, considerable variation in numerical characteristics, i.e., size and displacement, as well as in configuration, were found and are discussed in this Section. Figs. 6.38 and 6.39 shows the distribution of displacement and length of the three variants together with the multi-mission frigate from Test case 4 included for references.

Fig. 6.38 shows the AAW-frigate has the smallest displacement (3600 tonne), followed by the ocean-going patrol vessel (3900 tonne) and the ASW-frigate (4200 tonne), with the multi-mission frigate having the largest displacement (discussed previously in Fig. 6.9). Like other test cases, it is always possible to increase ship displacement with the same payload, hence the presence of large displacements up to and over 6000 tons for all four variants.

The differences in displacement are directly related to changes in dimensions. Fig. 6.39 shows the distribution in length of the variants. Again, the AAW-frigate has the smallest length (108 m), followed by the ocean-going patrol vessel (110 m) and the ASW-frigate (120 m), with the multi-mission frigate having the largest size. Again, it is always possible to increase ship size whilst retaining all systems, e.g., to improve sea-keeping behaviour (or for any of the other reasons outlined in Section 4.4.1).

The changes instigated by the payload variations are worthwhile to study from a configurational point of view, if only to gain insight in the differences in size and displacement shown in Fig. 6.38 and 6.39. Fig. 6.40 to 6.47 show two multi-mission frigates, two air-defence frigates, two anti-submarine frigates and two ocean-going patrol vessels, respectively. The multi-mission frigate configurations in Fig. 6.40 and 6.41 provide reference (and are repeated from Fig. 6.19 and 6.21). The differences are discussed below.

Several important observations follow from Fig. 6.40 to 6.47. Most obviously, the configurations differ in both hull size as well as superstructure shape confirming the observations from Fig. 6.39; they also have quite different internal arrangements. The configurations shown have a fairly traditional warship layout; more unconventional configurations were also found but are not included for reasons of brevity (some unconventional configurations were shown already in Fig. 6.16 to 6.23).

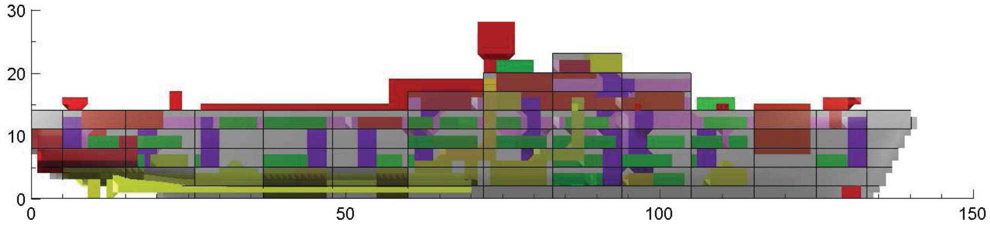


Fig. 6.40 – Multi-mission frigate: design no. 12964

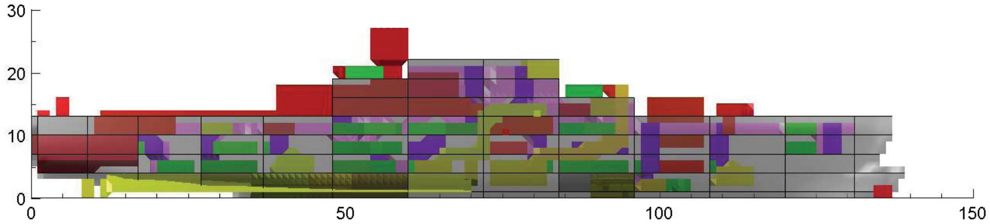


Fig. 6.41 – Multi-mission frigate: design no. 16748

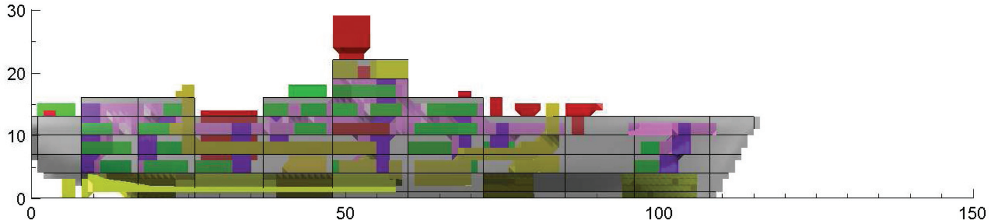


Fig. 6.42 – AAW-frigate: design no. 4200

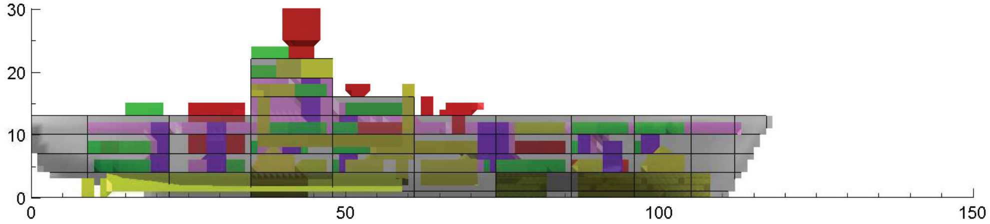


Fig. 6.43 – AAW-frigate: design no. 9120

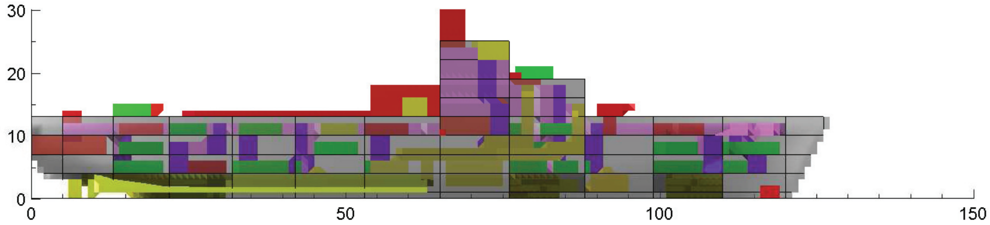


Fig. 6.44 – ASW-frigate: design no. 10686

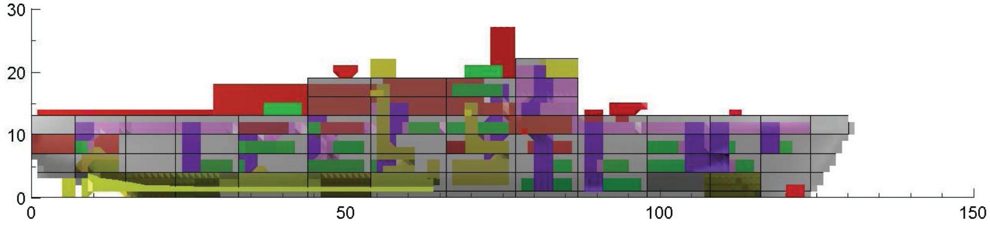


Fig. 6.45 – ASW-frigate: design no. 2591

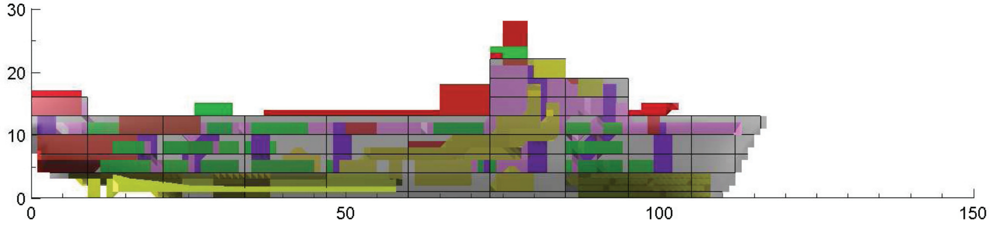


Fig. 6.46 – OPV: design no. 26298

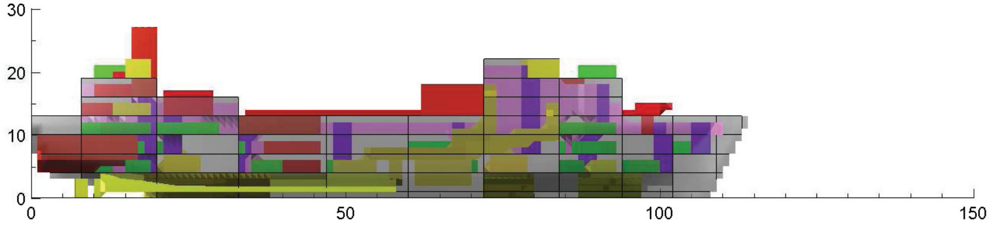


Fig. 6.47 – OPV: design no. 28923

The AAW-frigates in Fig. 6.42 and 6.43 have a very simple layout: a flat main deck with weapons and a small superstructure housing bridge, radar and uptakes. They are similar to the US Navy's Virginia class nuclear powered cruisers, in this respect. The lack of a flight deck, hangar and small boats located on the main deck both reduces ship length and reduces superstructure size compared to the multi-mission frigates in Fig. 6.46 and 6.47. The two configurations have their superstructure just aft of amidships to lower wave-induced accelerations in the operational areas.

The ASW-frigates in Fig. 6.44 and 6.45 are both slightly bigger and have a larger superstructure to house a helicopter hangar. One configuration (in 6.44) has a flight deck a third of the length forward of the stern, a position with lower vertical accelerations. The other, in Fig. 6.45 has a flight deck at the stern and her overall layout of the configuration is rather similar to the Royal Netherlands Navy M-class frigate (exception is the VLS, which is located next to the gun in Fig. 6.45). This illustrates how the packing approach can find conventional configurations (Fig. 6.45), in addition to more unusually configurations (like the those in Fig. 6.17).

The ocean-going patrol vessels in Fig. 6.46 and 6.47 are smaller in size than the ASW-frigate. The configuration in Fig. 6.46 has a large work deck aft harbouring the flight deck, RHIBS and twenty-foot containers, similar to many offshore supply vessels. The configuration in Fig. 6.47 differs. It has the flight deck amidships, with a twin-island superstructure. The aft island houses the radar and RHIBS; the forward island encloses bridge, uptakes and downtakes.

Summarised, Fig. 6.40 to 6.47 show that the packing approach can create coherent and feasible ship configurations for each payload variation listed in Table 6.7. Moreover, it is able to estimate impact on ship size (see Section 6.4 for a discussion of the accuracy of such estimates) as well as generate a range of different configurations for use by the selection approach discussed in Chapter 7.

Summary

The eleven test-cases illustrate the ability of the packing approach to generate coherent warship configurations that comply with a basic set of non-negotiable requirements, i.e., comply with overlap rules, float upright in stable condition with limited trim, and meet speed and endurance requirements.

Section 6.5 uses the results to discuss the issues outlined in Section 6.1. Before doing so, Section 6.3 first introduces the other application of the packing approach and search algorithm: deep water drillships.

6.3 Drillship application

6.3.1 Overview

Section 6.2 discussed the most elaborate application so far of the packing approach, though, it was not the first application. This Section therefore discusses the very first application of the packing approach: the early stage design of deep water drillships. The application was carried out by Karel Wagner as a MSc. research project. His project formed part of the PhD research presented here and was supervised by the author of this dissertation. The project was performed at the offshore and ship design firm GustoMSC in Schiedam, The Netherlands. The results are reported in *Wagner [135]*, *Wagner et al. [136]* and *Wagner et al. [137]*.

This Section gives a brief overview of the drillship application to investigate those issues outlined in Section 6.1 that were not considered in the warship design application in Section 6.2. Furthermore, this Section does not provide a discussion on drillship design. Five test cases were used in total to investigate various aspects of drillship configuration design. They are listed in Table 6.8.

Test case	Population size (-)	Generations (-)	Total designs (-)	Infeasible designs (-)	Feasible designs (-)	Feasible designs (%)	Reason for test case
1	120	42	5040	2441	2599	52	Riser hold aft of drilling assembly
2	120	50	6000	2803	3197	53	Engine rooms positioned in foreship
3	100	87	8700	4405	4295	49	Added 2 DP thrusters
4	100	79	7900	4554	3346	42	Accommodation aft
5	100	112	11200	5278	5922	53	Riser hold aft and engine rooms forward

Table 6.8 – Overview of five drillship test cases (from *Wagner [135]*)

The five test cases again used a single ship description. Changes were brought about by altering parameter values and, for Test case 4, by altering packing sequences. Section 6.3.2 briefly covers the parametric model, the systems of objects, packing sequences and parameters, while Section 6.3.3 discusses the results of two of the test cases: no. 4 and 5.

6.3.2 Parametric ship description

Approach used to develop the description

The approach used to develop the drillship model is similar to that followed for the warship application in Section 6.2. Starting point is a functional decomposition, followed by system allocation; the clustering of systems into objects and their integration in a parametric ship description. See *Wagner [135]* and *Wagner et al. [137]* for a more detailed discussion. A proprietary GustoMSC drillship design: the P10000NG, provided data on systems' sizes, weight and connectivity.

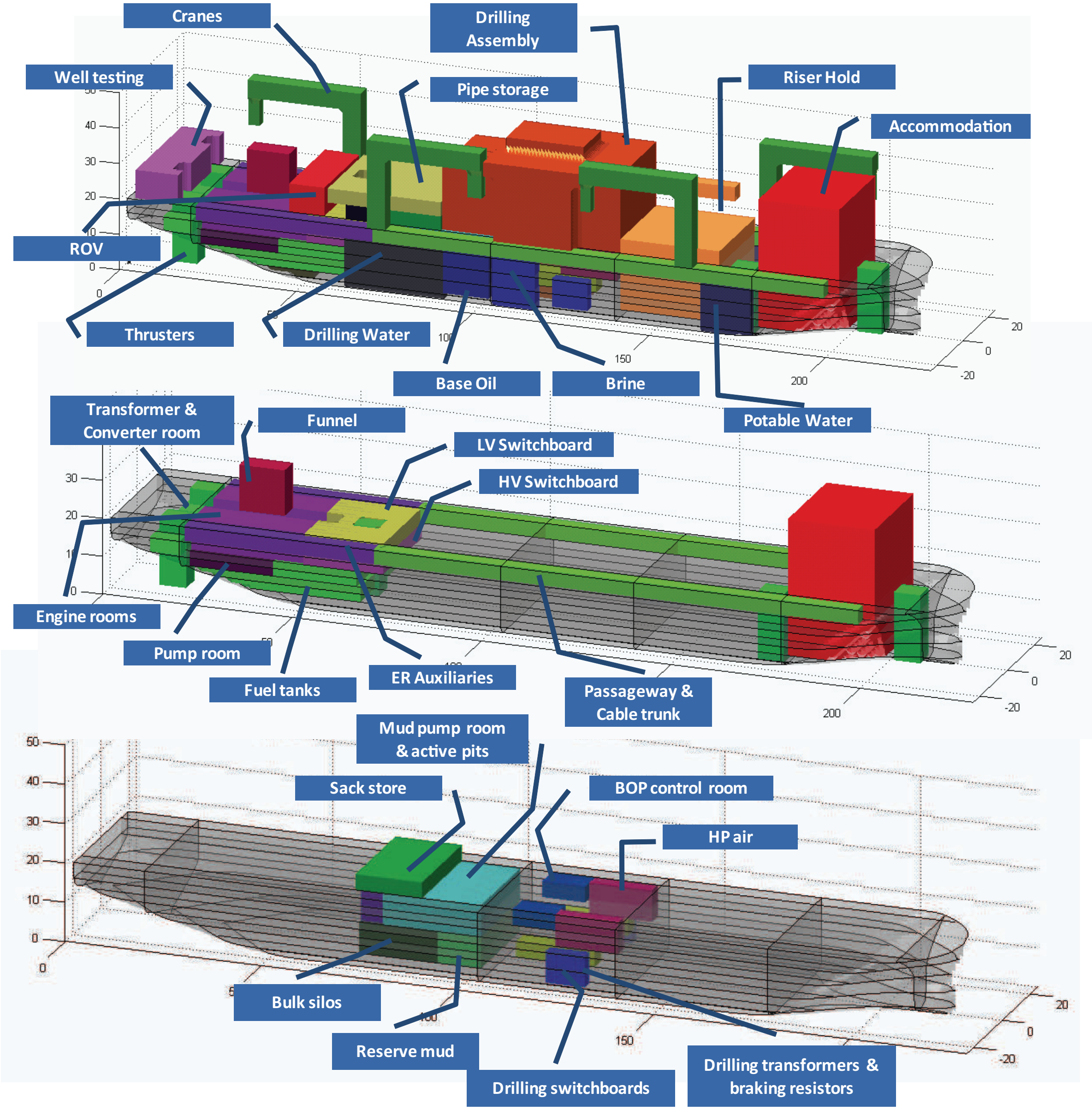


Fig. 6.48 – Systems used in the parametric drillship description (from *Wagner [135]*)

Systems of objects

The parametric ship description uses 50 systems of objects in total. Fig. 6.48 illustrates them using a representative drillship configuration. The upper image shows the drilling-related systems of objects, the middle the ship-related systems, the lower the auxiliary drilling systems. The systems of objects are also listed in Figs. 6.49 and 6.50. The parametric ship description did not use the bulkhead positioning approach outlined in Section 4.4.5 since reserve buoyancy requirements for civilian ships are less severe than those for warships. Instead, bulkheads are placed individually together with other large objects. A full discussion of all systems of objects and the systems they represent is found in *Wagner [135]*.

The parametric drillship description developed by *Wagner [135]* uses a different approach to focussing than the warship description used in Section 6.2. For example, it made little use of logical objects for focussing (discussed in Section 4.6). Instead, it relied heavily on constrained longitudinal and transverse positions as well as fixed initial positions to control the configuration, resulting in a more heavily constrained ship description. This increased the yield of the search process considerably (compare Table 6.8 to 6.1), at the cost of reducing diversity. Section 6.4 considers this in more detail.

Packing sequence

The packing sequence is shown in Figs. 6.49 and 6.50. It broadly follows the sequence outlined earlier in Section 4.5. Main difference is the positioning of individual bulkheads attached to large systems, such as accommodation and drilling derrick. Packing sequence for Test case 4 differed slightly to alter the position of the accommodation (again, see *Wagner [135]*).

Objectives & constraints

The drillship application used two objectives in the search process (both discussed in Section 5.4), namely packing density and proximity. The latter uses a user-defined weighting to ensure relevant pairs of objects become positioned in close proximity during the search process. The disadvantage of this is the lack of diversity that may result from it. Section 6.4.2 considers the impact.

The constraints that were used, were similar to those outlined in Section 5.4. Main difference was the lack of a constraint for reserve buoyancy, for reasons outlined earlier. Eight loading conditions were considered in the ballasting prediction (see *Wagner [135]* for their definition) and handled using the constraints on stability, trim and buoyancy in Section 5.4. Note that the ability of the packing approach and search algorithm to satisfy constraints from Section 5.4 is not considered in this Section, as it was investigated already in Section 6.2.

6.3.3 Results & discussion

Only two test cases are discussed in this thesis, namely Test cases four and five, which illustrate the applicability of the packing approach to drillship design in general, as well as its ability to cope with forced changes in configuration (see *Wagner [135]* for a discussion of all five test cases). The results from the two test cases are again discussed from a numerical and a spatial point of view, similar to the warship application presented in Section 6.2. Numerical characteristics are discussed first, using Fig. 6.51 as the starting point.

```
1 envelope
2 decks
3 dummy
   1 hard
4 dummy
   1 hard
5 bulkheads
6 accommodation
   1 soft (volume)
7 bulkheads update
8 drilling assembly
   1 hard
   2 hard
   3 hard
   4 and_not
9 bulkheads update
10 thruster FWD
   1 hard
   2 free space
   3 or
11 thruster AFT
   1 hard
   2 free space
   3 hard
   4 free space
   5 or
12 conv trans
   1 soft (area)
13 engine rooms
   1 hard
   2 and_not
   3 and_not
14 funnel
   1 or
   2 hard
15 thruster AFT FWD
   1 hard
   2 free space
   3 hard
   4 free space
16 thruster FWD AFT
   1 hard
   2 free space
17 ER auxiliaries
   1 or
   2 soft (area)
18 ER pump room
   1 soft (area)
19 riser hold
   1 hard
   2 free space
   3 shafts
20 bulkheads update
21 passageway
   1 hard
22 bulk silos
   1 hard
```

Fig. 6.49 – Systems 1 to 22 of the configurational model

```
23 APV
   1 hard
24 drilling SWBD
   1 hard
25 drilling trans brake
   1 hard
26 ROV
   1 hard
   2 free space
   3 soft (area)
27 FO tank 1
   1 soft (volume)
28 FO tank 2
   1 soft (volume)
29 FO tank 3
   1 soft (volume)
30 SWBD HV
   1 soft (area)
31 SWBD LV
   1 soft (area)
32 mud pump pits
   1 soft (area)
33 sack store
   1 soft (area)
34 pipe handling crane SB AFT
   1 hard
35 pipe handling crane SB FWD
   1 hard
36 pipe handling crane PS AFT
   1 hard
37 pipe handling crane PS FWD
   1 hard
38 cement unit
   1 hard
39 BOP control
   1 soft (area)
40 reserve mud
   1 soft (volume)
41 store 1
   1 soft (area)
42 pipe storage
   1 soft (area)
43 well testing
   1 soft (area)
44 potable water tanks
   1 soft (volume)
45 brine tanks
   1 soft (volume)
46 base oil tanks
   1 soft (volume)
47 drill water tank
   1 soft (volume)
48 dirty oil
   1 soft (volume)
49 bilge tank
   1 soft (volume)
50 bulkheads update
```

Fig. 6.50 – Systems 23 to 50 of the configurational model

Numerical characteristics

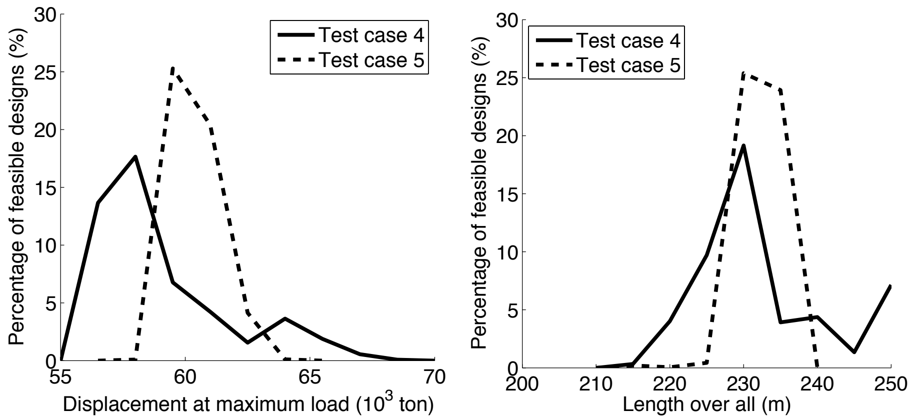


Fig. 6.51 – Distribution of drillship displacement (left) and length (right)

Fig. 6.51 shows the distribution of displacement and length of the two test cases. Both test cases result in quite large ships, especially compared to the warship discussed previously. Nonetheless, the range of displacements includes that of the original P10000NG's displacement (roughly 65.000 tonne), with some configurations being smaller, while others are larger. Furthermore, there is a considerable difference between Test case 4, with the accommodation placed at the stern of the ship, and Test case 5, with the accommodation at the traditional position at the bow of the ship. Differences in length are similar to the differences in displacement, the configurations of Test case 4 are considerably shorter. No in-depth investigation of the causes of this is presented here, see *Wagner [135]* for a discussion of the results. Fig. 6.51 does show how the packing process results in a range of alternative configurations.

Configurational characteristics

Configurational aspects are also of interest and Fig. 6.52 to 6.55 show four of the drillship configurations. See Fig. 6.48 to relate colours to system names. A quick comparison reveals differences in hull size and fullness (compare Fig. 6.53 to 6.55), as well as considerable differences and similarities in the positions of objects inside and on the ship. Main similarities are the drilling derrick, placed roughly amidships and the position of dynamic positioning thrusters (near bow and stern). Main differences concern most other objects. Again, a detailed discussion of configurations is not provided here, see *Wagner [135]* for details.

One last interesting observation. Aside from the search runs in Table 6.9, *Wagner [135]* also made a replica of the P10000NG by manually fixing initial positions of objects. Most interestingly, Test case 5 found a configuration, shown in Fig. 6.55, very similar to this replica. The characteristics of both configurations are shown in Table 6.9 (see *Wagner [135]* for a more detailed comparison).

	P10.000NG replica	Design no. 9230 Test case 5	Unit
Length over all	230	230	[m]
Maximum beam	36	36	[m]
Depth to main deck	18	18	[m]
Max. displacement	60717	63137	[t]
Max. draught	10.4	10.7	[m]
Cb^2	0.69	0.69	[-]
L/B	6.4	6.4	[-]
B/T	3.5	3.4	[-]
Minimum GM	1.00	1.00	[m]
Minimum freeboard	7.6	7.3	[m]
Total lightship weight	30795	30795	[t]

Table 6.9 – Comparison of a P10000NG replica and design no. 9230 from Test case 5

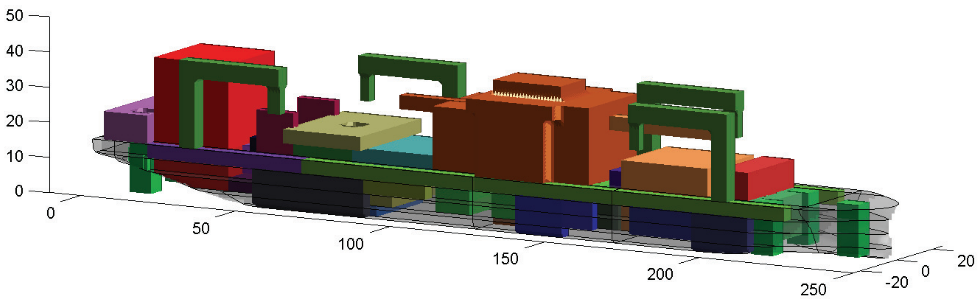


Fig. 6.52 – Configuration no. 1839 (Test case 4)

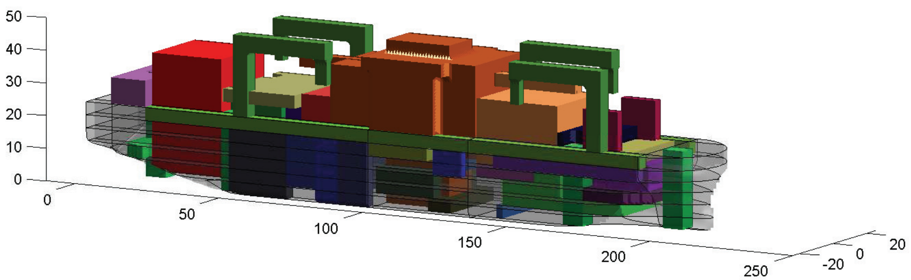


Fig. 6.53 – Configuration no. 2276 (Test case 4)

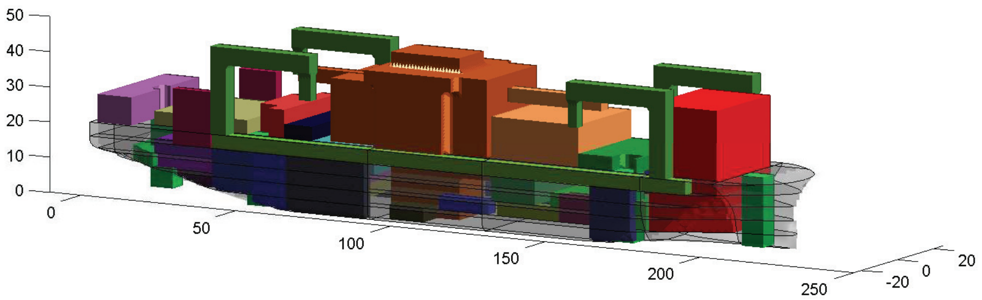


Fig. 6.54 – Configuration no. 558 (Test case 5)

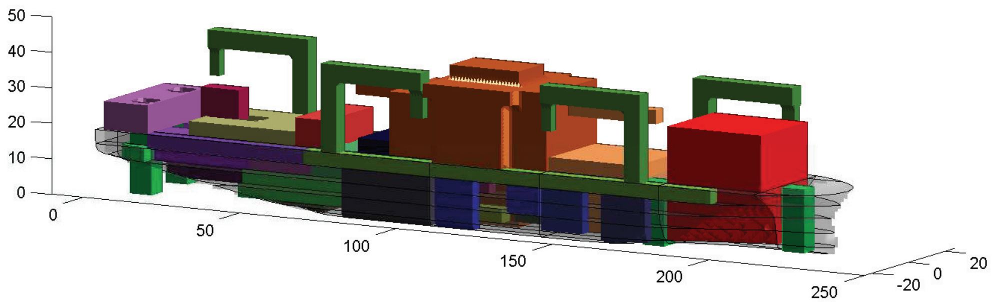


Fig. 6.55 – Configuration no. 9230 (Test case 5)

Summary

Though deliberately brief, the discussion of the results of Test cases 4 and 5 illustrate the packing approach is able to successfully describe a complex civilian service vessel and generate a range of alternative drillship configurations that differ both in displacement and length. The search runs also found a configuration very similar to the P10000NG used as a parent ship. Section 6.5 combines the results from the drillship application with those from Section 6.2 to reflect upon the issues laid out in Section 6.1. Before doing so, Section 6.4 will first investigate some theoretical issues concerning the packing approach.

6.4 Some theoretical issues

6.4.1 Introduction

The previous Sections showed two applications of the packing approach and illustrate how it successfully creates a diverse set of feasible ship configurations that comply with a basic set of non-negotiable design requirements. It is deemed appropriate to investigate three specific theoretical issues to obtain a better understanding of the packing approach's ability to find coherent ship configurations. These issues are:

- The impact of constraints on main hull dimensions.
- The relation between constraining and diversity.
- The reliability of the size estimates.

The three issues are respectively considered in Sections 6.4.2, 6.4.3 and 6.4.4.

6.4.2 Ability to search: impact of constraints on main dimensions

The first test case of the warship application in Section 6.2 has a limited range of main dimensions (see Fig. 6.10 and 6.11).

One possible cause of this limited variation in size could be that the ship has to be quite large in order to accommodate all systems. Alternatively, NSGA-II could fail to consider a sufficiently wide range of dimensions during the search process. The latter would be particularly troublesome, as there is little use for the packing-based ship description in early stage ship design if it is unable to vary ship dimensions.

The cause of the limited variation of main dimensions needs to be determined. There are three possible causes of the limited variations in main dimensions: the user-defined bounds on main dimensions; the impact of constraints safeguarding non-negotiable requirements; and an unexpected inability of NSGA-II to vary main dimensions during the search process.

The investigation uses the main dimensions, i.e., length, beam and design draught, of all configurations considered during the search process regardless of feasibility. The set is filtered to remove those configurations that do not meet particular constraints to reveal the effect on main dimensions. Filtering considers constraints in a fixed sequence, due to the fixed packing sequence (Section 4.5) and the staggered prediction of performances (Section 5.5). The sequence of filtering is outlined below; Figs.6.56 and 6.57 show the effect of the three constraints on length, beam and draught. They also show the bounds on main dimensions (indicated by a dashed line) for length, beam and draught.

The use of contour plots and the resulting use of the contouring algorithm in *Matlab* seems to have slightly increased the range of dimensions shown (especially on the bounds of the envelope dimensions). Nonetheless, the observations presented below were unaffected by this slight increase.

1. **All designs**. First to be considered are the dimensions of all configurations evaluated by NSGA-II during the search process within bounds on envelope dimensions. The lightest shade in Figs. 6.56 and 6.57 represents these configurations. It shows nearly the full range of dimensions within the user-defined bounds were considered during the search process.

2. **Designs with all systems placed**. First constraint is the ability to place all systems of objects while complying with overlap rules defined in Chapter 4. This provides a lower bound on the minimum ship size required to accommodate all systems. The next darker shade in Figs.6.56 and 6.57 represents this group and shows how the requirement to accommodate all systems of objects increase minimum length and beam considerably. The minimum length that complies is 115 m (increased from 100 m for category 1) , minimum beam increased from 12 to 14 m, while the range of design draughts remains unchanged.

   These changes can be expected, as increasing length and beam increases the available space inside the envelope, space necessary to accommodate all systems. The lack of changes in design draught is also expected, as the draught only concerns the required buoyancy (which is not considered by this constraint, but in the next constraint, which will be discussed below) and has little impact on the space available inside the envelope (which is governed by the height to main deck and superstructure size).

3. **Designs with all systems placed, and sufficient buoyancy**. Being sufficiently big is not enough, the ship also needs to float. Adding an extra buoyancy check (not listed in Section 5.4.2) to the previous constraint on overlap compliance makes it possible to identify the minimum ship size that has enough buoyancy to carry the weight of the ship (without using water ballast to adjust for trim and stability). Adding the requirement to float increases minimum length and beam (to 120 m and 15 m respectively). The range of design draughts does not change as, apparently, the minimum length and beam required to accommodate all objects, result in an envelope that is also sufficiently big to provide enough buoyancy.

4. **Feasible designs**. The final set of configurations are those that comply with all constraints in Section 5.4.2, i.e., their size is big enough to accommodate all objects, they float upright with appropriate trim and proper initial stability (while using water ballast -where necessary- to meet buoyancy, trim and stability constraints). The requirements on initial stability and trim have a major impact on length, beam and draught. Minimum length and beam again increase, to respectively 130 m and 17.5 m.

   The reason why is that some ships are necessarily longer to make it easier to meet the near-zero trim requirement. Also, the minimum beam of the ships increases to improve initial stability up to required level. The changes in design draught are also of interest, as maximum design draught is reduced. Probable cause is a lack of stability. A large design draught that far exceeds the actual draught of the vessel reduces the water-line area considerably, with a reduction of initial stability as a result. Water ballast could be used to ballast down the vessel to a stable draught, however, this requires internal space for ballast tanks, which may not be available. No investigation was made, though it would be fairly straightforward to carry out by considering differences between design draught and actual draught and plotting it against the initial stability at the actual draught.

   Note that the ballasting calculation considers trim, buoyancy and initial stability concurrently. As such, it is impossible to determine the individual impact of constraints on trim, stability and buoyancy.

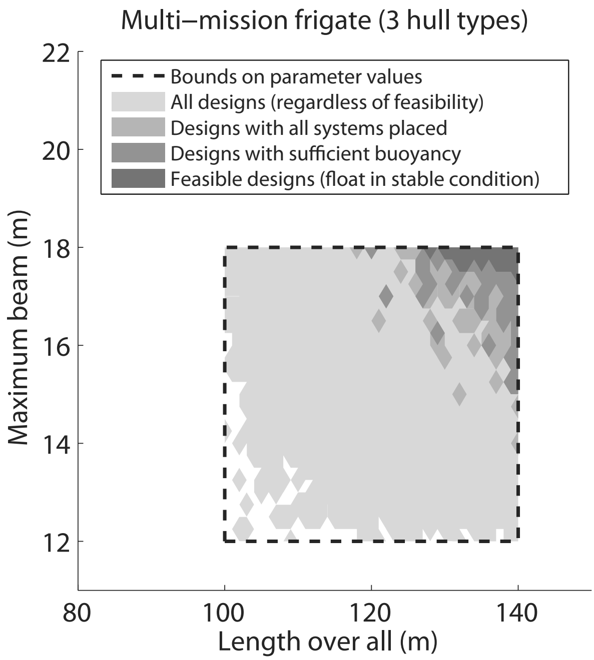


Fig. 6.56 – Impact of feasibility requirements on length and beam (frigate application, test case 1)

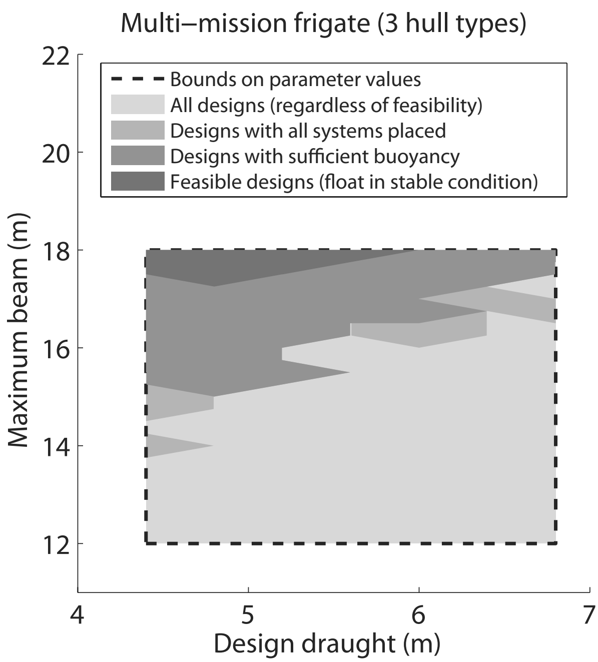


Fig. 6.57 – Impact of feasibility requirements on beam and draught of a frigate (frigate application, test case 1)

Several important conclusions can be drawn from Fig. 6.56 and 6.57.

- First, there is no inability to consider the full range of main dimensions during the search process. Instead, the limited variation in ship size it is caused by bounds on main dimensions and the feasibility requirements. Loosing either of them (preferably the bounds on dimensions) will result in further variation in dimensions.

  Note, Fig. 6.56 and 6.57 vindicate the choice made in Section 4.5 to pack the envelope as the very first object. As such, they illustrate that the full range of dimensions was considered during the search process, despite using a 'classic' packing approach to handle the interaction between envelope and other objects. Hence, the use of 'packing' over 'wrapping' (the latter proposed by *Andrews [3]*) does not need to be restrictive.

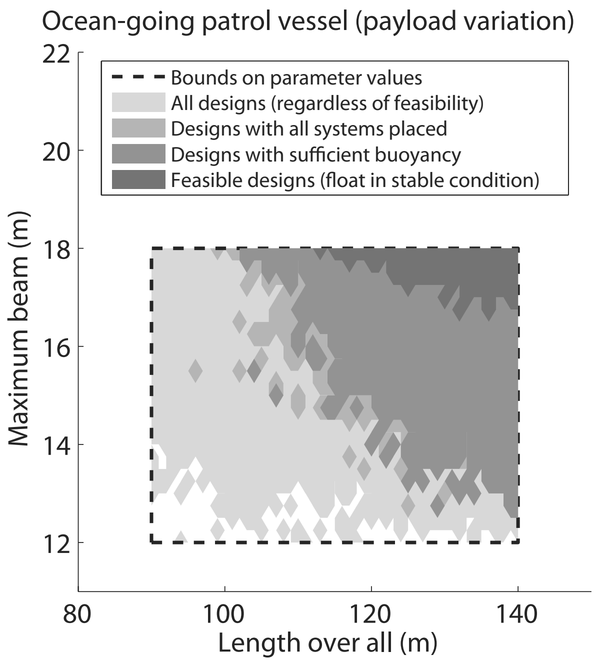


Fig. 6.58 – Impact of feasibility requirements on length and beam of the ship (frigate application, OPV-variant from Test case 11))

  This is illustrated further by Fig. 6.58. It shows the effects of the constraints on the range of lengths and beams for the ocean-going patrol vessel discussed in Section 6.2.3 (see Figs. 6.46 and 6.47, for instance). The reduction in the number of systems (Table 6.7) reduces the required minimum length, and therefore effectively loosens the upper bound on main dimensions. The range of main dimensions of the feasible designs increases as a result, as shown in Fig. 6.58, which illustrates that constraints and bounds on dimensions together cause the limited variation of main dimensions in Fig. 6.56.

- Second, it is much easier to find large ship configurations that comply with the feasibility requirements, than it is to find smaller configurations. As a result, the upper bounds on length and beam are almost always considered during the search process. This again strengthens the decision taken in Sections 5.4.1 and 5.4.3 to use packing density as an objective to pursue for more and more dense configurations. Without this objective, the ship configurations would be even more likely to have the maximum allowed dimensions.

6.4.3 Diversity

The configurational model for the frigate differed from the drillship application in the methods and degree of constraining and, hence, their resulting yields. An example is shown in Fig. 6.59 and 6.60. They show the use of prohibited x-positions that constrain positions of particular systems of objects. A black dot indicates a prohibited position. The drillship model (in Fig. 6.59) uses considerably more constraining than the frigate model in Fig. 6.60.

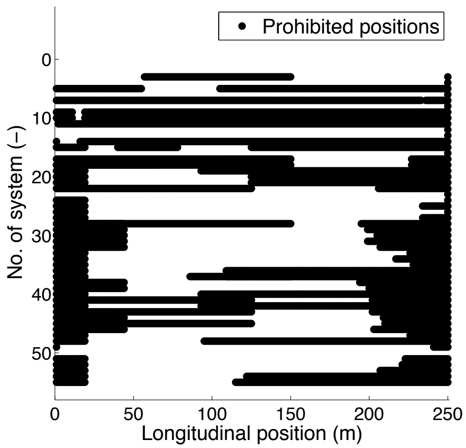


Fig. 6.59 – Degree of longitudinal constraining per system for the drillship application

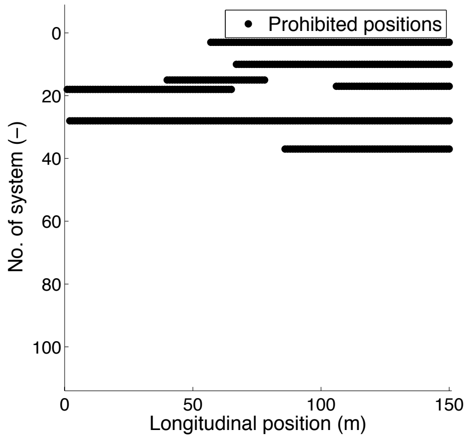


Fig. 6.60 – Degree of longitudinal constraining per system for the frigate application

This raises the question as to what extent the degree, and ways of focussing, affects diversity. To study whether diversity actually changes, a comparison is made between two search runs: the AAW-frigate (Section 6.2.3, e.g., shown in Fig. 6.42) and the drillship (Test case 5, discussed in Section 6.3.3, e.g., shown in Fig. 6.52). The comparison uses the diversity measure discussed in Section 5.3 to calculate the diversity of two sets of configurations. First, the mean diversity of all feasible configurations in the current child population is calculated. Second, the diversity of the most diverse configuration in the current child population. Both diversity measures are calculated for every generation, resulting in the plots in Fig. 6.61 and 6.62.

The differences between the AAW-frigate and the drillship are striking. The diversity in the AAW-frigate is near constant during the entire search process, whereas the drillship reduces almost instantly after the first feasible configurations are found. In addition, the diversity of the AAW-frigate is considerably higher than that of the drillship test case. Both differences illustrate that the frigate configurational model is able to create and maintain a more diverse set of configurations, at a cost of lower yield (compare Table 6.1 to 6.8). The impact of the considerable use of symmetry in the drillship model was considered separately by removing the transverse system position from the diversity calculations.

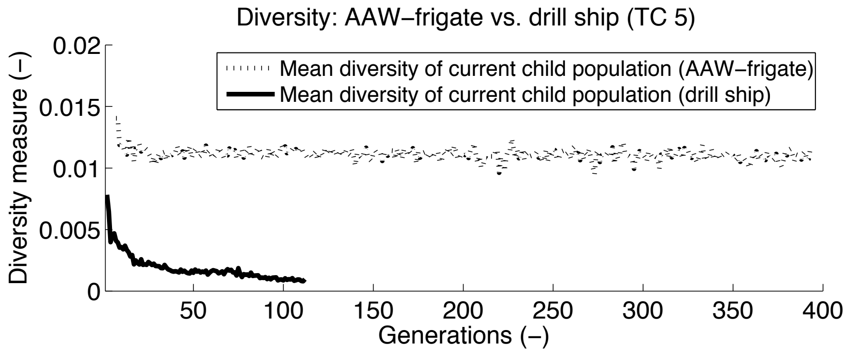


Fig. 6.61 – Mean diversity in current child population: drillship vs. AAW-frigate

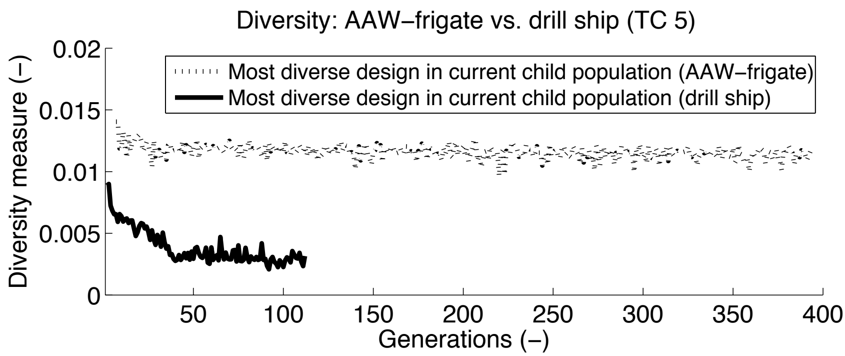


Fig. 6.62 – Most diverse design in current child population: drillship vs. AAW-frigate

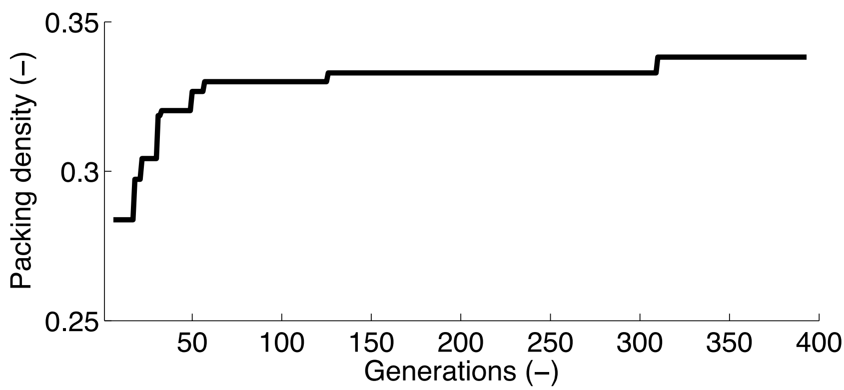


Fig. 6.63 – Convergence of packing density for the AAW-frigate

One might argue that the near constant diversity of the frigate application observed in Fig. 6.61 and 6.62 is caused by a lack of convergence of the packing density objective (defined by Eq. 5.6 in Section 5.4.3). Fig. 6.63, however, shows that this is not the case. The maximum packing density steadily increases throughout the search process, without any apparent effect on diversity.

Summarised, the configurational model for the frigate from Section 6.2 is indeed less restrictive than that used for the drillship application from Section 6.3. More specifically, reducing the constraints on systems positions (Fig. 6.60) and expanding the use of logical objects (Section 4.6) seems to promote diversity while maintaining coherency of configuration. As a result, the diversity of the feasible designs remains nearly constant while the search process converges towards more dense designs.

6.4.4 The reliability of the size estimates

The search runs in Sections 6.2.3 and 6.3.3 result in a large number of feasible configurations. This set was used to investigate the impact of different requirements on the main dimensions of the ship. This Section provides some thoughts on how reliable these size estimates are.

Deviations in the size estimate can have two consequences: some errors might lead to an overestimate of ship size while other errors might lead to an underestimate of ship size. Both deviations are entirely possible (and sometimes occur in combination) for different reasons.

- **Ship size can reduce further**. Section 3.4.2 explained search algorithms use a finite number of design iterations to generate ship configurations. Given the vast number of combinations of parameter values, it is therefore unlikely that the smallest configuration will be considered during the search process. As such, it is very well possible the minimum ship size found during the search process is an over-estimation, and may well be reduced further. Only a detailed investigation of the configuration will reveal the extent to which further size reductions are possible. As such, the size of the smallest configuration found must be considered as a relatively conservative, i.e., high, estimate of minimum size, which can then be used as a starting point for further design efforts.

- **Ship size must increase**. Sections 3.4.2 and 3.4.3 explained the lack of completeness of ratings search algorithms face during the search process. As a result, configurations that meet all constraints may nonetheless retain basic flaws, flaws that could be removed by considering sub-optimal feasible configurations, e.g., that are bigger.

  For example, using only a reserve buoyancy calculation (Section 5.3), but not a damaged stability prediction, may result in configurations which do float when damaged to design limits but have insufficient stability to remain upright. This lack of stability could be addressed by altering the configuration, e.g., by increasing beam. This does increase ship size, however, most likely in a manner such that the new ship dimensions become larger than those of the smallest configuration found during the search process.

  Fortunately, one can select a bigger configuration to start with (using the selection approach presented in Chapter 7) in an attempt to prevent such size increases, which is possible due to availability of a large set of configurations that covers a range of sizes.

Whether size reductions and size increases cancel each other out is impossible to say beforehand. Detailed inspection of the chosen configurations and the additional performance predictions required should help establish this (the refinement step in Fig. 3.6). As such, one should always be aware that both reductions and increases in size may prove necessary, when interpreting the results in Sections 6.2.3 and 6.3.3.

Curiously, naval architects may face exactly the same questions in their day-to-day practice: 'can the ship become smaller?' and 'does it need to become bigger?'. Most importantly, they also face exactly the same challenges as search algorithms: more time and design iterations needed to make the ship smaller, and more knowledge and performance predictions to figure out if it needs to be bigger. In general, using the parametric ship description to build a replicate of an existing ship and predict its performances helps to gain part of the insight required (as, for instance, as did *Wagner [135]*).

6.5 Conclusions: is the packing-based approach suited for early stage ship design?

This Chapter applied the packing-based ship description developed in Chapters 4 and used it in combination with the search algorithm from Chapter 5 to generate large, diverse and feasible sets of frigates (Section 6.2) and deep water drillships (Section 6.3). What remains is to use the results presented in this Chapter to discuss and reflect upon the issues laid out in Section 6.1. They are repeated and answered below.

- Issue 1: *Describe the entire ship configuration at variable level of detail, one deemed suitable by the naval architect.* The entire ship configuration is described using the object types introduced in Chapter 4, i.e., hull, superstructure, decks, bulkheads and systems and their connections. Moreover, level of detail is adjustable, as a comparison between the drillship application in Section 6.3 and the frigate application in Section 6.2 reveals (Figs. 6.16 to 6.25 versus 6.52 to 6.55).

- Issue 2: *Instigate large changes to the ship configuration by changing parameter values.* The parametric ship description uses a range of input parameters to change the initial shape and position of systems of objects. Among these changes were large changes in envelope shape (see Fig. 6.20 and Fig. 6.21), induced by a discrete input parameter, as well as envelope dimensions, with length ranging between 90 and 140 m and width ranging between 12 and 18 m (see Figs. 6.56 and 6.57, for instance). Input parameters also changed the positions of a large number of objects leading to large changes in configuration (compare Figs. 6.18 to 6.24). Hence, the large changes are indeed induced by changing input parameter values.

- Issue 3: *Apply these changes concurrently to the entire ship configuration, i.e., hull, superstructure, decks, bulkheads, systems and their connections.* Sections 4.8, 6.2 and 6.3 illustrate the packing approach is indeed able to apply large changes concurrently to the entire configuration, while maintaining a coherent design (e.g., Fig. 6.19 to 6.23). This was achieved thanks to overlap management (outlined in Section 4.4.5) to change the parameter-driven initial shapes and positions of objects to ensure all meet the overlap rules in Section 4.4.3. Most importantly, the frigate application varied three hull types concurrently during the search process (Fig. 6.9) to study the impact on the resulting configurations.

- Issue 4: *Allow variable degree of focusing by the naval architect.* The frigate application in Section 6.2 and drillship application in Section 6.3 use both different means and degrees of focussing (see Sections 4.6, 6.2.2 and 6.3.2). Together, the applications illustrates how the same packing approach, with the same overlap management and overlap rules handles different degrees of focussing, with a more or less diverse set of configurations (Section 6.4.3) and different yields (Table 6.1 and 6.8) as a result.

- Issue 5: *To study whether the same packing approach can be used for different ship types.* The frigate application in Section 6.2 and drillship application in Section 6.3 use the same implementation of the packing approach. This illustrates the packing approach is indeed able to model at two different types of service vessels with the same object types, overlap rules and design changes for overlap management (though the systems that the object types represented obviously differed considerably between the two applications!).

- Issue 6: *Use the packing approach to generate a large and diverse set of feasible ship configurations.* The applications presented in Sections 6.2 and 6.3 resulted, respectively, in roughly 6800 different frigate configurations and approximately 19300 different drillship configurations that all satisfy the constraints in Section 5.4.2. Diversity concerns dimensions and displacement (see Fig. 6.9 to 6.12 and 6.51) as well as spatial aspects (see Fig. 6.16 to 6.25 and 6.52 to 6.55). The investigation whether configurations satisfy the constraints was performed in Section 6.2.3 (see Fig. 6.13 to 6.15).

- Issue 7: *Assess whether the use of assumptions outlined in Section 5.5 ensures the correct speed, range and reserve buoyancy.* Section 5.5 mentioned the use of assumptions, i.e., assumed draught, in this instance, to predict required propulsion power and reserve buoyancy. The investigation in Section 6.2.3 reveals the design draught is not exceeded, thus safeguarding the predictions that were based on the assumed draught. Furthermore, one can choose the margin between actual draught and design draught to be either larger or smaller (Fig. 6.13), e.g., the former can help hedge against increases in total ship displacement caused by erroneous weight estimates.

- Issue 8: *Assess whether the approach is able to investigate changes in requirements that can reasonably be expected to be considered during the early stage design of service vessels.* The frigate application looked into the variation of design requirements. More specifically, variations of performance requirements, e.g., range at cruise speed (Fig. 6.34 and 6.37) and separation of redundant systems (Fig. 6.32 and 6.33) were made, as were variations in hull shape (Fig. 6.30 and 6.31) and weapons and sensor fit (Fig. 6.38 to 6.39). Though far from exhaustive, the results do illustrate the packing approach is responsive to such changes and, therefore, should be able to help investigate the many options facing the naval architect (Section 1.3).

The results presented in this Chapter (and summarised in the discussion of the eight issues identified in Section 6.1) illustrate the packing-based ship description and search algorithm together can generate a large set of different ship configurations that meet a set of non-negotiable requirements. Moreover, the packing approach is considered being reusable -to a large extent, as it can handle two quite different types of service vessels. Lastly, and perhaps most importantly, the packing approach can handle several examples of the large variations in requirements likely to be faced during the design of service vessels, i.e., both changes in performance requirements (e.g., range, in Fig. 6.34), as well as changes to the systems the ship must carry (Fig. 6.38 to 6.47).

Chapter 7

Selecting ship configurations

7.1 Introduction

The previous three Chapters discussed the packing approach, the search algorithm and the application of them in two test cases. What remains is the last step of the process outlined in Section 3.6 and repeated in Fig. 7.1, namely: the selection of a few, promising alternatives that represent a suitable compromise between negotiable requirements.

The selection approach's development builds upon the discussions in Sections 3.4.2, 3.4.3, 3.5.2 and 3.5.3 that, respectively, identified and proposed remedies for a range of issues encountered during and after the use of search algorithms. Moreover, the examples that illustrate its development will use the results from the frigate application presented in Chapter 6.

A rough outline of the selection approach was presented in Section 3.6. Before developing this outline further, the required capabilities are stated below, as they form the aims of the development effort. They will be used in Section 7.4 to assess the capability of the selection approach.

- **Provide transparency**. All relevant characteristics of the entire set of configurations must be thoroughly investigated to learn why particular configurations are of interest and why others are not (Section 3.5.3).

- **Address the lack of completeness of ratings**. The number of objectives and constraints used to guide the search process is finite and, therefore, additional means to distinguish between designs and to identify flaws are necessary (Section 3.5.3). The expression of engineering judgement triggered by visualising configurations in the proper context is seen to provide this means.

- **Resolve the implicit nature of compromise decisions**. Selecting designs means taking compromise decisions to settle trade-offs between conflicting design requirements. When doing so, a selection decision remains provisional until all its relevant consequences have become known, due to its implicit nature (Section 3.5.3). To resolve this implicit nature, one must be able to rapidly take, consider and revise selection decisions.

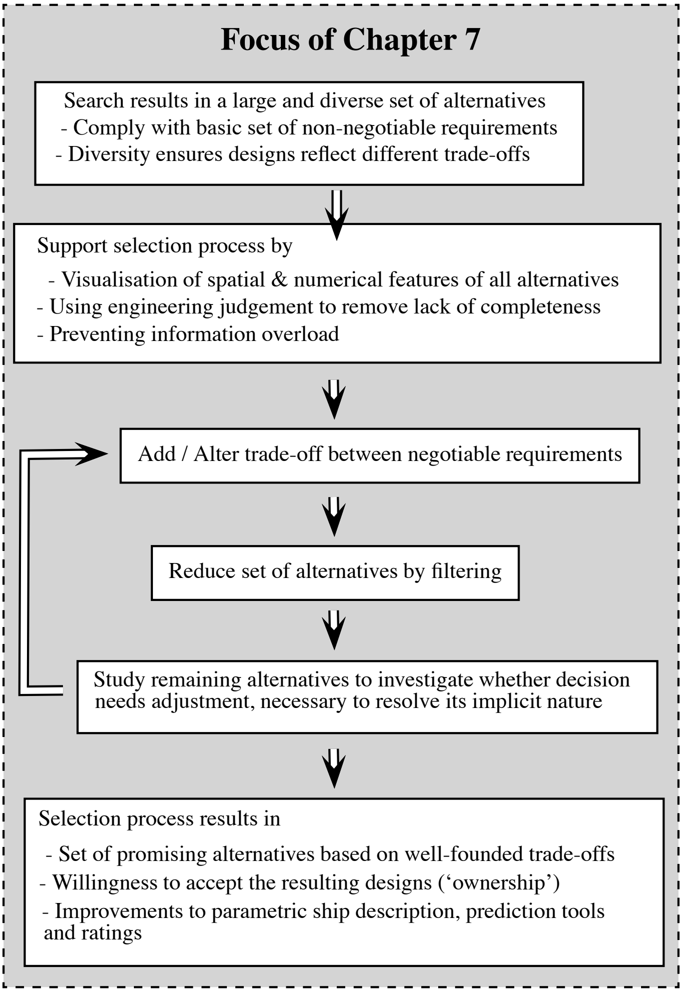


Fig. 7.1 – Focus of this Chapter

- **Generate acceptance**. Last issue to be addressed is acceptance, i.e., the willingness of naval architects to use configurations generated by the search algorithm and packing approach and that are chosen during the selection process. Without acceptance, results from the search effort will not be used and will have been generated in vain.

The development of the selection approach starts in Section 7.2 by discussing a simple selection process to identify the steps taken during selection of promising designs. Section 7.3 develops the approach in detail using these steps. Section 7.4 subsequently applies the selection approach to choose a promising air defense frigate configuration; an example upon which Section 7.5 will reflect. Section 7.6 concludes this Chapter by reviewing the extent with which the selection approach is able to provide the four capabilities outline above.

7.2 A generic selection process

Before presenting the selection approach in Section 7.3 it is worthwhile to consider a simple selection process to study the steps to be taken to reduce a large set of configurations to a single promising alternative. The steps are derived by running through a generic selection process, which is discussed below.

Obviously, the most important step for the selection process is generating a set of configurations that meets the non-negotiable design requirements outlined in Chapter 5, to relieve the naval architect from the burden of maintaining these non-negotiable requirements, and allow him or her to focus on identifying designs representing promising trade-offs. Fortunately, Chapter 6 illustrated that the packing approach (Chapter 4), combined with the search algorithm (Chapter 5), can generate the required set of configurations.

Preferably, the set of configurations should be both large and diverse to enable the naval architect to investigate a wide range of trade-offs. Whether such a set can be generated depends, among other things, on the parametric ship description and the constraints used. For example, Section 6.4.2 illustrated that the range of dimensions for the feasible designs can be quite small, despite relatively large bounds on main dimensions.

Other steps to select a promising configuration are introduced below. Note, the introduction of these steps does not include examples. For such examples, please see the more detailed discussion in Section 7.3.2.

1. **Selecting is deciding**. Starting point is the essence of selection: it is deciding which designs in a set will be retained and which designs are not liked and thus can be discarded. Only then can a set of configurations be reduced.

2. **Decision-making is a human responsibility**. The naval architect is responsible for taking the selection decision, for two reasons. Firstly, decisions are human inventions (Section 2.4) and secondly, making humans responsible for decision-making instills acceptance for the outcome, i.e., generates a sense of ownership, without which no naval architect who bears design responsibility will accept the outcome of the selection process (again, Section 2.4).

3. **Selecting requires multiple selection decisions**. Selecting a suitable configuration from a large set rarely occurs in a single decision, for two reasons.

   Firstly, the boundary between desirable and undesirable options is not clear cut. Usually coarse decisions are taken first (selecting groups), and more refined decisions later (selecting individual designs); a procedure which requires multiple decisions.

   Secondly, different characteristics should be considered during the selection process. For example, identify affordable configurations first by selecting those below a user-defined cost ceiling, before considering those designs with a superior operability in a particular wave climate. Again, considering different characteristics requires taking multiple decisions.

   Therefore, selecting configurations requires multiple selection decisions, which are taken in a sequential manner, until the larger set of configurations reduces to a handful of designs.

4. **Deciding requires insight**. Properly deciding which designs to retain requires the naval architect to have insight why to retain or discard them. Decisions might be made without such insight, however these tend not to be the best decisions. The degree with which decisions are subject to revision is discussed below (see 'selection decisions are implicit' on page 221).

5. **Insight requires transparency**. Gaining insight requires transparency, i.e., the ability to investigate all relevant characteristics of all feasible configurations, regardless of their ratings. Moreover, correlations between characteristics should be identifiable to help the naval architect recognise the need for trade-offs.

6. **Transparency requires visualisation**. Visualisation helps to provide transparency by both plotting numerical results, e.g., size, displacement, or speed in scatter plots, as well as spatial aspects, such as the positions of systems.

7. **Transparency requires filtering**. Investigating all relevant characteristics of all feasible configurations means considering a vast amount of information. Therefore, filtering is required to prevent information overload. Filtering is provided by considering only a small subset of characteristics at a time, and change the set of characteristics sequentially. Several filters could also be combined to study their effect.

   Filtering and visualising spatial aspects is more problematic and requires a bespoke visualisation approach, as, for instance, the concurrent visualisation of a thousand general arrangement plans is incomprehensible to humans.

8. **Insight requires addressing the lack of completeness**. Selection decisions cannot be based on insight gained by results that include 'too optimal', i.e., erroneous, configurations. Identifying and removing such designs requires addressing the lack of completeness caused by the finite number of numerical ratings and performance predictions used during the search process.

9. **The lack of completeness can be addressed by using engineering judgement**. Addressing the lack of completeness can be achieved through the application of human judgement to distinguish between designs. This requires the investigation of all relevant characteristics of all feasible configurations. The reason is that the expression of human judgement to rate designs on aspects not considered during the search process may render designs that are 'sub-optimal' in the formal sense (that have less than optimal values for the rule-based objectives considered during the search process) 'optimal' in the practical sense of the word (i.e., deemed preferable by the naval architect). See Section 3.5.2.

10. **Expressing engineering judgement (related to ship configurations) requires the consideration of spatial features in the proper context**. Getting naval architects to express engineering judgement is difficult; they must be triggered to do so. To help trigger expression, the selection approach should confront the naval architect with the relevant data on the ship designs, i.e., present the information he or she needs to express engineering judgement.

    What constitutes 'relevant data' depends on the topic one wants to express engineering judgement about, e.g., propeller characteristics, waves-induced motions, or diesel engine characteristics. Each area of expertise in naval architecture uses different data presented in different ways. For example, expressing judgement on propeller design could be facilitated by confronting the naval architect with parameters such as a thrust coefficient, a torque coefficient and an advance coefficient.

Since the lack of completeness or ratings identified in Chapter 3 concerns the spatial configuration of the ship in general, and the consequences performances have on it in particular, the 'relevant data' in this Section are considered to be the spatial features of the ship designs, i.e., the positions and shapes of systems.

11. **Expressing engineering judgement requires proper context**. Expressing engineering judgement is helped by providing the proper context to present the relevant data in. For example, one might plot the longitudinal positions of a flight deck and bridge in a simple scatter plot, but this does not offer the right context. Instead, when these two positions are plotted in a general arrangement drawing (used in any ship design project), they become more meaningful.

12. **Selection decisions are implicit**. The availability of the set of feasible configurations enables the naval architect to focus on settling trade-offs (Section 3.5.2). This means selection decisions become predominantly trade-off decisions and, as a result, selection decisions inherit their implicit nature.

    Furthermore, the use of filtering to prevent information overload (discussed above), means it is often not clear what all relevant consequences of a particular selection decision will be (after all, filtering means one considers only a few aspects at a time), which strengthens the implicit nature of selection decisions.

13. **Resolving implicitness of decisions requires working out all relevant consequences**. The implicit nature of selection decisions means they may have unwanted consequences, which may cause them to be revised. As such, all relevant consequences of a decision: intended and unintended, must be worked out by continuing the selection process until a single configuration remains.

14. **Resolving implicitness requires ability to reconsider and revise decisions**. Once all relevant consequences of a selection decision have become known, it becomes necessary to decide whether or not to revise the selection decision. Whether a particular selection decision was indeed appropriate, or whether it needs adjustment should then become clear. More specifically, it helps to establish whether priorities (i.e., human opinions that deem one thing more important than other things), need adjustment.

15. **Revising decisions must be fast**. Lastly, the ability to reconsider decisions is meaningless if it cannot be performed in a rapid manner. Only then can insight be gained into the suitability of selection decisions, the trade-offs they settle, and the underlying priorities on which the decisions are based. Fortunately, the availability of a large set of configurations with predicted performances helps to maintain the speed of decision-making, since the reversal of selection decisions concerns a change in filtering of the designs from the database.

Together, the steps outlined above should help the naval architect select an acceptable and promising configuration from a larger set by taking numerous selection decisions themselves.

The selection is based on insight gained from a thorough and transparent investigation of both numerical and spatial features of all feasible configurations. Moreover, the lack of completeness can de addressed by rating configurations with engineering judgement, an approach that

```
While sufficient designs remain
    From high to low priority

        Select option 1, 2 or 3

        Option 1: select using numerical values
            Select numerical aspects to consider
            Retrieve and visualise available numerical values
            Select suitable numerical values by drawing a polygon

        Option 2: select using system position relative to main dimensions
            Select system to consider
            Retrieve and visualise available positions
            Select suitable positions by drawing a polygon

        Option 3: select using system position relative to another system
            Select system to use as reference
            Select system to consider relative to first system
            Retrieve and visualise available relative positions
            Select suitable relative positions by drawing a polygon

        Remove designs falling outside selection polygon
        Retain polygon for reference purposes
        Document reasons for selection

    End
End

Visualise remaining designs upon sufficient reduction
```

Fig. 7.2 – Outline of the selection process

is supported by presenting spatial features in a familiar context (a general arrangement drawing). Lastly, the selection decisions are implicit and therefore subject to revision. Removing their implicit nature is achieved by continuing the selection process until all relevant consequences become known, reviewing the decisions, and revising them if necessary in a rapid decide-consider-revise cycle.

In summary, the steps should help resolve the issues raised earlier in Sections 1.5, 2.6 and 3.4. Section 7.3 uses them to develop the selection approach in detail.

7.3 The selection approach

7.3.1 Outline of the selection approach

Section 7.2 discussed the steps of a generic selection process to properly select an acceptable and promising design from the large set of alternatives. What remains is their integration in the overall selection approach, which is outlined in Fig. 7.2.

The process in Fig. 7.2 is explained in more detail in Section 7.3.2 and includes a discussion how the steps from Section 7.2 are implemented.

7.3.2 Detailed explanation of the selection approach

Preparation: characteristics to consider and their priorities

The use of the selection approach starts with preparation. Aside from generating a set of feasible configurations (Chapter 6), there are a few other things to be done before using the selection approach.

First, it is necessary for the naval architect to identify those characteristics which are to be considered during the selection process, e.g., size of the ship, positions of important systems or wave-induced accelerations. As discussed, characteristics can be both spatial (position of a drill derrick) or numerical (the ship's length). Also, the naval architect can consider any characteristic of any feasible design, even those that require the expression of engineering judgement. Furthermore, the chosen characteristics can be adjusted at any time prior to the selection process. Equally important, the list does not have to be complete. However, knowing before what to consider during selection helps guide the selection process towards configurations of interest.

Second, characteristics will differ in importance. These differences are based on the naval architect's priorities and may vary among the persons involved in the design of service vessels. Different priorities result in different trade-offs between negotiable requirements and therefore different configurations. Priorities and their resulting trade-offs are implicit, however, and the key purpose of the selection process is figuring out whether the naval architect's priorities need adjustment. The first step towards assessing the need for adjustment is stating the priorities attributed by creating a sequence in which characteristics will be considered (ordered from high priority to low priority). The sequence is considered frozen during the selection process in Fig. 7.2, but can be changed and the selection process repeated to investigate different priorities.

The characteristics to be considered and their priorities are used during the investigation of the configurations.

Investigating configurations: filtering and visualisation

Once preparation finishes, one can start the investigation of all feasible configurations. First thing to be considered is the prevention of information overload by filtering (Section 3.5.3). Filtering is provided by presenting only a few characteristics of all feasible designs in a sequential manner. The sequence used for filtering is the one discussed previously; it runs from characteristics with high priority to those with low priority. The combination of considering of a few characteristics of interest at a time, and changing them sequentially, helps to keep the amount of information manageable and also enables one to consider all characteristics of interest, of all feasible configurations under consideration. As discussed, the sequence of filtering is considered frozen during the selection process, but can and should change in-between different selection runs.

The next step is that of visualisation. The selection approach in this Chapter visualises only two characteristics at the same time, using two-dimensional scatter plots. More advanced plot types, such as contour plots and carpet plots, can plot more characteristics at the same time (three and four, respectively) but were not used to simplify the development of the selection approach.

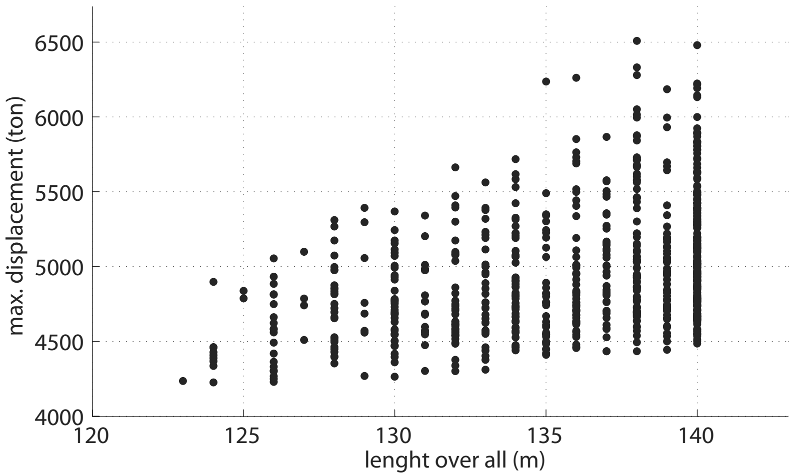


Fig. 7.3 – Scatter plot of numerical characteristics: displacement against length (771 ASW-frigate configurations in total)

Furthermore, the naval architect must be able to study both numerical and spatial features of all configurations using visualisation, whilst also being able to filter it. Both the visualisation of numerical features and the visualisation of spatial features are discussed below.

- **Visualising numerical features**. Visualising numerical features is straightforward and is achieved using a standard two-dimensional scatter plot as in Fig. 7.3, which displays the displacement and length of 771 feasible Anti-Submarine Warfare frigates (see Section 6.2.3 and Table 6.1). One can visualise and filter any numerical characteristic of configurations in this manner by sequentially adjusting the features to plot.

  Important drawback of numerical plots, such as in Fig. 7.3, is they do not support the expression of engineering judgement related to ship arrangements. For that, the selection approach needs to present spatial features of configurations, which is discussed below.

In comparison, visualising spatial features requires far more attention. The reason is that proper visualisation of spatial characteristics in the right context is considered to be the key to triggering the expression of engineering judgement related to ship arrangements (Section 3.5.3). Still, developing the visualisation requires another topic to be addressed: making the visualisation of spatial features filterable to prevent information overload.

Filtering, in the most generic sense, prevents information overload by presenting the naval architect with only part of the spatial information available. The ship's spatial description consists of a large but finite number of objects, each with both a shape and position. This means one can filter on either shape or position.

- **Filtering and visualising positions**. Filtering positions is fairly straightforward. The ship configuration is represented by a finite number of objects, each with its own position. This means one can filter positions by considering the positions of a small subset of objects, e.g., only the bridge (Fig. 7.4) or the flight deck and a RHIB (Fig. 7.5).

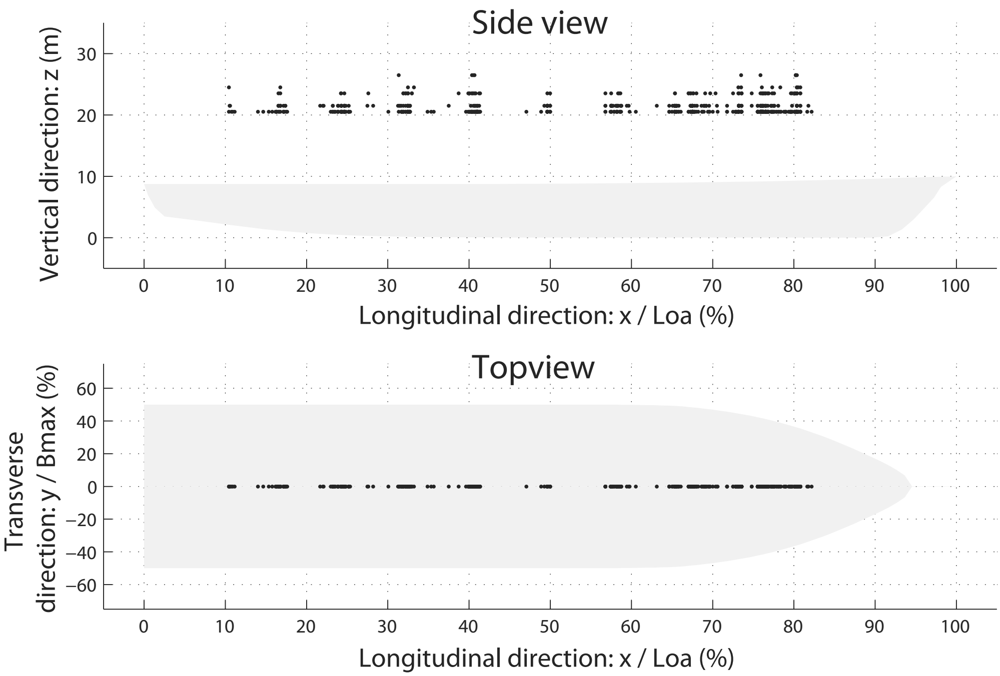


Fig. 7.4 – Position of the bridge relative to the envelope

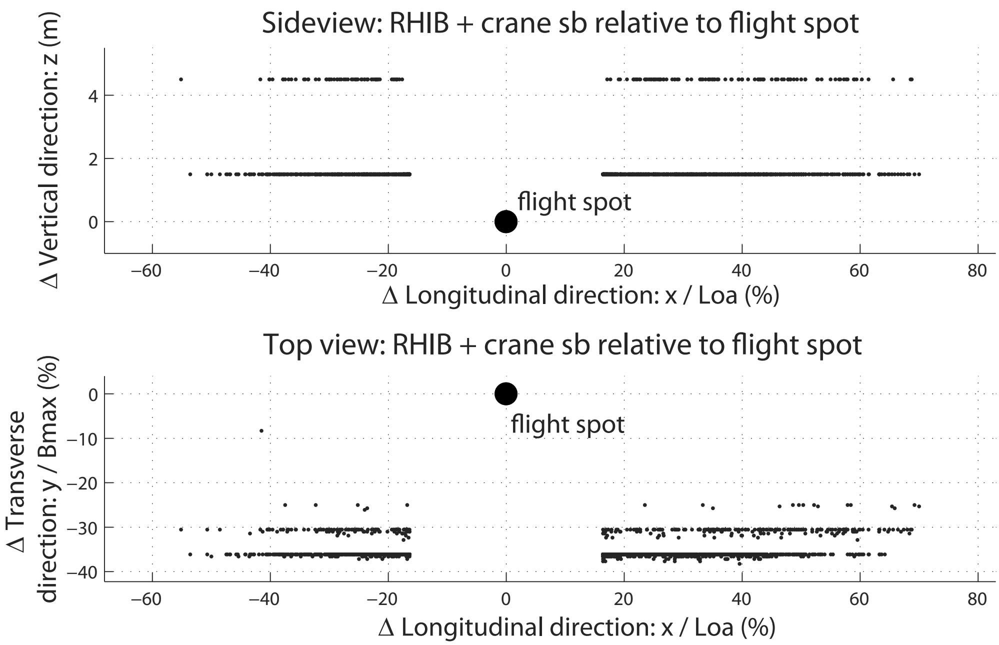


Fig. 7.5 – Position of the RHIB relative to the flight deck

Filtering is facilitated by the fact that an object's centre of gravity is a discrete point in a three-dimensional coordinate system. This enables one to plot the positions of a particular object for multiple configurations in a single image (e.g., the bridge in Fig. 7.4), without loosing insight.

Note, one can filter on the position of an object relative to the ship's envelope, e.g., determine whether the bridge should be near bow or stern (Fig. 7.4), as well as relative to another object, e.g., the position of a RHIB and crane relative to the flight deck (Fig. 7.5). Sequentially altering the objects under consideration enables one to cover the positions of all objects of interest. The x and y-coordinates are made dimensionless by dividing them with the length and beam of the ship, respectively. z-coordinates are not made dimensionless, since the shape and height of the superstructure will vary considerably and making them dimensionless would not give the naval architect an accurate impression on the actual vertical position of systems.

Since the packing approach can configure three-dimensional ship configurations, one requires two two-dimensional views: a side-view (upper image in Fig. 7.4) and a top-view (lower image in Fig. 7.4).

- **Filtering and visualising shapes**. Filtering shapes is anything but straightforward. The reason is that a shape has a distinct continuous nature, e.g. consider a line made up of an infinite number of points, quite unlike the distinct discrete nature of a position.

  This means visualising the shape of a particular object for numerous configurations becomes rapidly unclear. For example, the concurrent visualisation of fifteen triangles in Fig. 7.6 illustrates this. The triangles have different shapes but the same position (all have their centroid at $x = 0, y = 0$). Even for such a limited number of simple shapes does the concurrent visualisation of numerous shapes become difficult to comprehend.

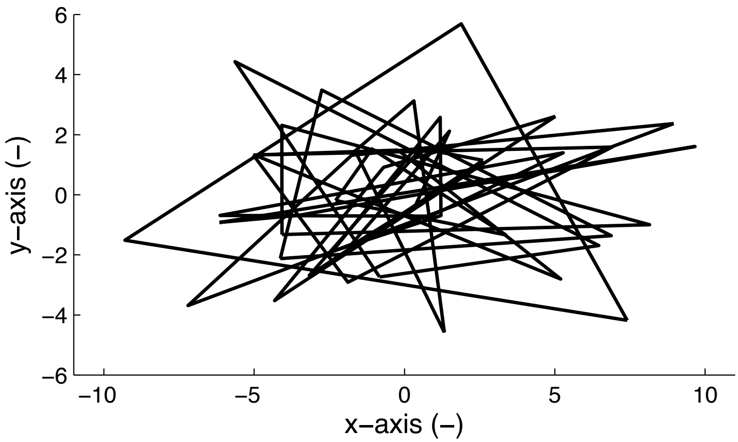


Fig. 7.6 – Concurrent visualisation of fifteen triangles (their centroids are at $x = 0, y = 0$)

One solution would be to discretise a shape into a set of points and use the same approach as used for filtering positions. Instead, an even simpler solution is used: object shapes will not be visualised during the selection process. The reason is that, generally speaking, shapes of most objects are considered to be of less interest during the selection process than positions.

> Detailed reasons vary. Shape is fixed for some objects, e.g., hard objects and free space objects (see Section 4.4.1 for details), while for other objects, the packing approach ensures they meet user-defined constraints (e.g., soft objects). For connection objects, one can derive shape from the objects used as starting and ending points (again Section 4.4.1), in combination with the knowledge that a shortest-path routing algorithm has been used[1]. The envelope is an important exception, as its shape is of obvious interest. Still, even envelope shape can be neglected to a reasonable extent, as the interaction between other object types and the envelope offers subtle information on the envelope's shape. For instance, selecting a bridge position amidships (Fig. 7.4) means one must have a superstructure underneath it to ensure support (following Section 4.7.2).

In summary, of the spatial features of a configuration, only positions will be visualised -using simple points- to indicate all available positions of a particular system in the set of feasible configurations. Both system positions relative to the envelope, and system positions relative to the position of other systems are visualised. The visualisation of spatial features is crucial for expressing engineering judgement related to ship configurations. This is discussed next.

Context and expressing engineering judgement

What has not been discussed so far is context. Positions could be visualised using simple scatter plots, e.g., similar to length and displacement in Fig. 7.3. However, scatter plots provide little insight how the numerical values for the x, y and z-positions relate to the overall ship configuration. Instead, the selection approach should use a context familiar to any naval architect: a general arrangement plan, to plot the positions in. For example, Fig. 7.4 shows the positions of the bridge relative to the envelope plotted in a side-view arrangement plan. It is much more meaningful than an ordinary scatter plot, as the positions in Fig. 7.4 already offer a glimpse of what the overall ship configuration could look like.

Fig. 7.7 will also serve as an example how the familiar context can be used to trigger the expression of engineering judgement. For instance, the image provides sufficient information

[1] However, not visualising shape for connection objects is not entirely satisfactory. For example, a passageway's shape many have been adjusted to get around other objects. Such a deviation that cannot be spotted by considering only the starting and ending points.

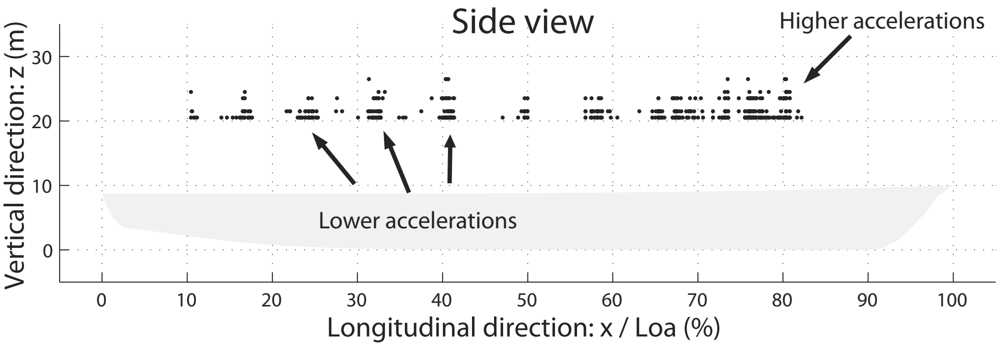


Fig. 7.7 – Bridge positions with higher and lower accelerations in waves

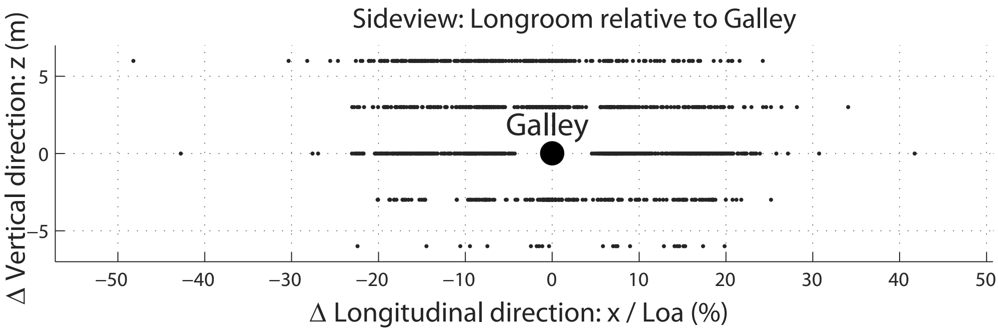


Fig. 7.8 – Relative position of the longroom relative to the galley

to any naval architect to identify those bridge positions that will have lower accelerations in a seaway (located at a longitudinal position between 20% to 40% of the length forward of the stern) and those that have higher acceleration (high up near the bow).

The visualisation of relative positions can be used in a similar manner. Fig. 7.8 shows the position of the longroom relative to the galley. It provides enough information to identify relative positions that are practical, e.g., having longroom and galley on the same deck in-between the same bulkheads helps transfer food, as well as rather less practical configurations, with a larger distance between the two.

Both examples illustrate how providing a suitable context facilitates the expression of engineering judgement. Triggering its expression is a crucial element of the approach outlined in Section 3.6. When successful, as Fig. 7.7 and 7.8 show, it enables the use of a very small set of ratings to guide the search process towards a diverse set of configurations. This eliminates the need to formulate ratings to capture trade-offs between requirements, something argued to be not worth the effort in Section 3.5.2.

Still, visualising and filtering is only the first step of the process in Fig. 7.1, one must also select preferable configurations, which is discussed below.

Selecting: 'hand-drawn' polygons to enclose desirable configurations

Visualisations such as Fig. 7.3 and 7.4 show, respectively, all available numerical values and all available positions of a particular system, in the set of feasible configurations. Not all values are preferable, though, and one needs the means to indicate which designs to retain.

A very simple approach is used to express the selection decision. One draws -using a mouse- a closed polygon around the values or positions deemed preferable. The polygon is simple and intuitively to create. It can have any shape or size, is suited to select any group of numerical values or object positions, and can therefore be used to select any desirable feature of a configuration (apart from shape, for reasons outlined earlier). The use of polygons is beneficial, as their creation is both simple and fast, and thus suited for the rapid investigation of different priorities (Section 7.2).

An example is shown in Fig. 7.9. A simple rectangle encloses those bridge positions deemed desirable, e.g., for the operability reasons shown earlier in Fig. 7.7. Again, the naval architect

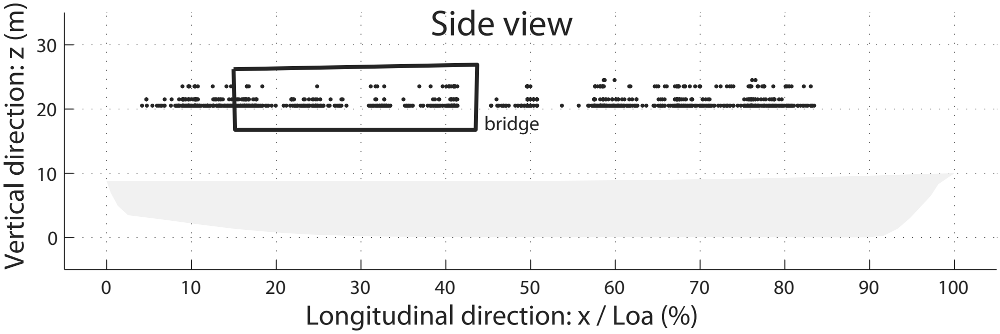


Fig. 7.9 – Enclosing preferable bridge positions with a selection polygon

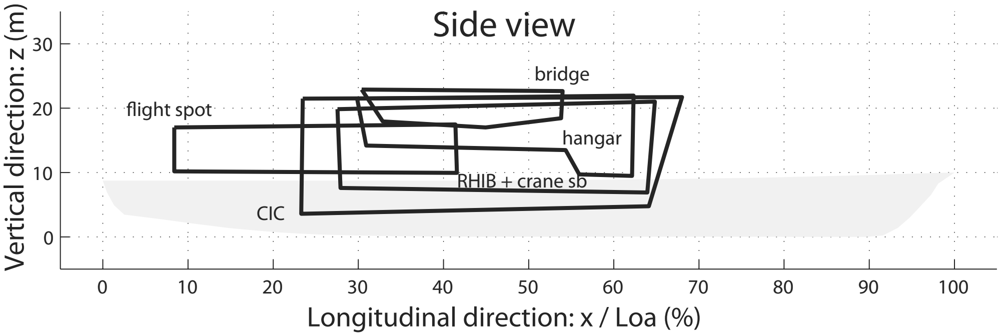


Fig. 7.10 – Polygons identifying previous selection decisions

is free to draw a polygon with any shape at any position deemed appropriate. Those designs that do not comply with the selection decision lie outside the polygon (note, in Fig. 7.9 the designs outside the polygon are designs that according to Fig. 7.7 should experience higher wave-induced accelerations).

Section 7.2 explained one does not select a promising configuration with a single decision. Instead, one uses multiple decisions concerning multiple features of the design. One must thus use multiple selection polygons, as shown in Fig. 7.10.

Lastly, polygons used for taking selection decisions are not discarded. Instead, they are retained as a reference frame, i.e., one considers the polygons from earlier decisions when taking subsequent selection decisions. For example, one could use the polygon defining the hangar position in Fig. 7.10 as a reference when choosing a suitable bridge position.

Documenting decisions

Taking decisions reduces the set of configurations. Obviously, these decisions must be documented by storing the underlying rationale on which it was based. This is especially important as one should repeat the selection process in Fig. 7.2 numerous times to investigate different trade-offs using different selection decisions. Without storing arguments for decisions, one will forget why a particular decision was taken, hampering the comparison of promising alternatives after the selection process.

The ability of the selection approach developed in this Chapter to trigger the expression of human engineering judgement means it should be possible to capture such knowledge (as proposed in *van Oers et al. [130]*). This perceived ability was investigated and realised in a follow-on PhD-research project at Delft University. Presented in *DeNucci et al. [35]*, *DeNucci et al. [36]* and *DeNucci and Hopman [34]*, the approach enables designs to be generated using the packing approach from Chapter 4 to purposely trigger the expression of engineering judgement on particular features of a design to capture knowledge about it.

The formal capture of such knowledge helps to reduce the effort required to formulate the ratings search algorithms use to guide the search process. Efficient knowledge capture could thus help to improve yield, as the search algorithm should be more aware of what to search for (see Section 3.4.2). However, the application of captured knowledge for taking compromise decisions warrants attention, though, due to their implicit nature (again, Section 3.4.2).

Reducing the set of designs

The selection process in Fig. 7.2 starts with all feasible configurations. Once a selection polygon is drawn, it is used to determine which position of systems or values lie inside it and which outside. This is achieved using a simple mathematical procedure: the '*inpolygon*' function in *Matlab* (see help documentation of *The Mathworks Inc. [114]*), which returns a list of those positions that are enclosed.

By linking the system positions back to the individual configuration in the set, one can identify those configurations that do not comply with the selection decision. Those designs outside the polygon then are removed from the set of configurations that is considered during the selection process. This process of taking selection decisions by drawing a polygon and removing those designs located outside it is repeated until the set of configurations is reduced up to the point where only a few designs remain.

Investigating trade-offs

The detailed explanation of the process in Fig. 7.2 does not yet cover the investigation of different trade-offs. For this, selection decisions and / or sequence of decisions must be altered (the latter by attributing different priorities to the characteristics of interest).

A simple example will help illustrate the process, using the set of ocean-going patrol vessel configurations from the frigate test case in Chapter 6 (Table 6.1, Fig. 6.46, Fig. 6.47 and Fig. 6.58). Fig. 7.11 and 7.12 show two sets of selection decisions and the resulting configurations. The decisions concern the positions of flight deck and the Rigid Hull Inflatable Boats (RHIBs).

The flight deck is shown as the long flat object, the RHIB position as a higher, shorter rectangle. The configuration in Fig. 7.11 has the flight deck near the stern, with the RHIB position further forward adjacent to the hangar. The configuration in Fig. 7.12 differs considerably; the RHIB positions are near the stern and the flight deck has moved forward. Even though only two systems are considered, one can already use them to identify two conflicts between negotiable requirements:

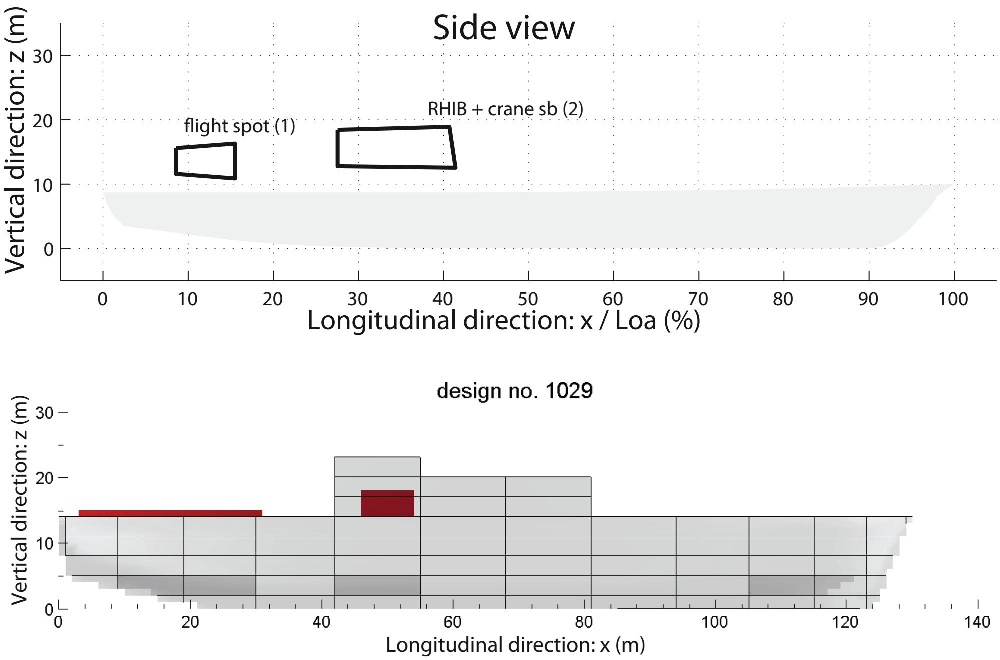


Fig. 7.11 – First set of decisions (upper image) and resulting configuration (lower image)

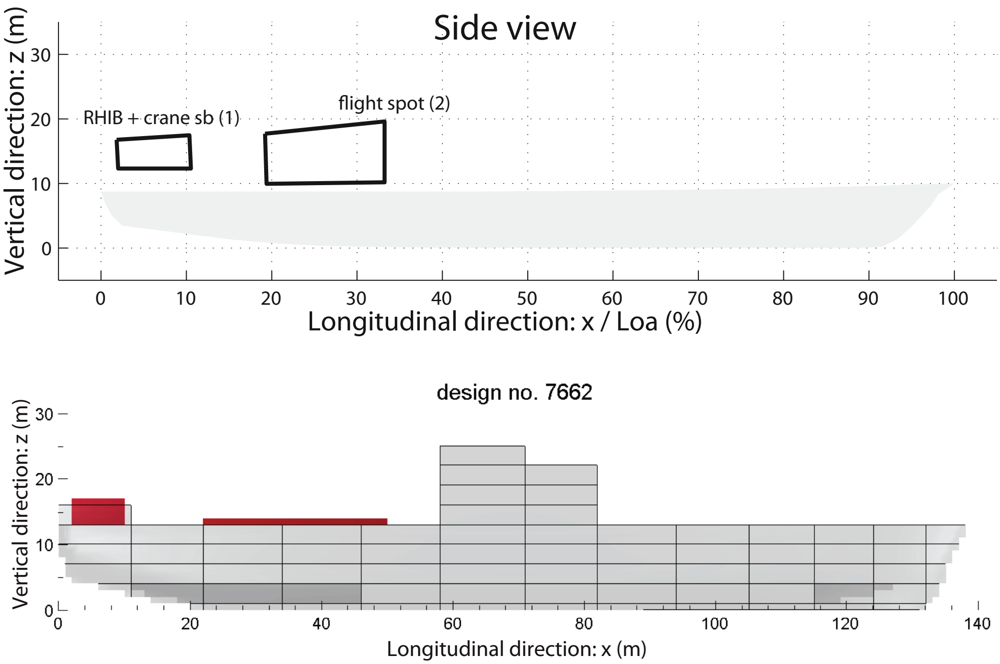


Fig. 7.12 – Second set of decisions (upper image) and resulting configuration (lower image)

- Sea-keeping. Both helicopter operations and small boat operations are influenced by ship motions. Assuming the relevant motions are indeed lower near 20 to 40 percent of the ship's length forward of the stern, one has to choose whether to place the flight deck or RHIB at this place. The configurations in Fig. 7.11 and 7.12 reflect the two different choices one has.

  Obviously, a more detailed prediction of the ship's sea-keeping performance and more carefully formulated operability criteria will offer a better foundation for differentiating between the alternatives in Fig. 7.11 and 7.12. Nonetheless, given the absence of such predictions, the expression of engineering judgement in the context provided helps making an initial decision between the alternatives.

- Concurrency of operation. One of the missions an OPV may perform is maritime security operations, a mission which may require concurrent operation of both helicopter and RHIB. With the configuration in Fig. 7.12, the RHIB crew may need to cross the flight deck to reach the RHIB prior to launching it, whereas configuration in Fig. 7.11 does not require this, as the RHIBs are located forward of the flight deck. As such, the configuration in Fig. 7.11 offers better support for concurrent RHIB and helicopter operations if one prefers such concurrency of operation. Note, this problem disappears if a RHIB launched via the stern ramp is used, but this is, obviously, not an option in the example being considered.

The two examples illustrate how the effect of different trade-offs on the ship configuration can be investigated by taking different selection decisions. What was not discussed is the consequences on overall ship size. The reason for this is that, firstly, one needs to add more selection decisions to arrive at a single configuration before one can firmly establish ship size. Secondly, the configurations were relatively spacious, meaning there was some freedom to alter the configuration without increasing the size of the ship.

Closure

The investigation of trade-offs discussed above completes the detailed discussion of the selection approach shown in Fig. 7.2. Section 7.4 will apply the selection approach to identify and select several promising Air Defence Frigate designs.

7.4 Applying the selection approach

Introduction

This Section uses the selection approach developed in Sections 7.2 and 7.3 to choose several promising concepts from the set of AAW-frigates discussed in Chapter 6. It will also discuss the underlying decisions and their rationale in some detail. Reasons to use the AAW-frigate is that the search run yielded 1407 feasible designs (Table 6.1). This number should provide ample alternatives to choose from.

The main purpose of this Section is to illustrate how the selection approach offers the naval architect the opportunity to investigate different trade-offs. Subsequently, he or she can use the insight to select the single configuration deemed most suitable. This Section also shows how the naval architect can select configurations based on both numerical and configurational aspects. Most importantly, the use of the selection approach will highlight the ability to include engineering judgement during the investigation and selection of designs. Section 3.5.2 argued this ability is crucial to address the lack of completeness caused by the inability to define all relevant ratings prior to performing a search run with NSGA-II.

The selection of the promising AAW-frigates is presented below. The investigation and the selection decisions outlined are relatively simple to show the principle. A more elaborate application would consist around dozens of investigations, each consisting of a dozen or so selection decisions.

Selecting promising concepts for an AAW-frigate

The test case with the AAW-frigate has resulted in 1407 alternative configurations (Table 6.1). To select several alternatives representing different trade-offs, it is necessary to decide upon what trade-offs and therefore which alternatives are of interest. For AAW-frigates, several interesting features come to mind (remember that the payload of all 1407 alternatives is the same, as are range, endurance, reserve buoyancy and the degree of separation between redundant systems).

Three features of interest are considered in the example:

- **Height of the radar above the water-line**. This determines the effective radar range, i.e., the distance between the horizon and the ship. In turn, this determines the time available to enact self-defence measures in the event of an attack by aircraft or anti-ship missiles.
- **Limits on concurrent use of weapon systems by overlapping firing arcs**. A 3-inch gun firing over a VLS means the two systems cannot be used at the same time. Whether this is acceptable depends on the kind of threats the ship is expected to encounter as well as their presumed concurrency. If possible, such overlaps should be avoided.
- **Crew fatigue by wave-induced motions**. The AAW-frigate does not operate helicopters or small boats and the weapon-systems tend to be less sensitive to ship motions (perhaps with the exception of the 3-inch gun). This means that the ship's configuration can be focussed towards improving crew operability in operational areas.

The three items listed above will be considered against the size and displacement of the configurations. Though not strictly proportional to costs, size and displacement can be used as an additional criterion for a simple reason: the payload of all configurations is equal, which means that smaller ships could be more cost-effective (thought the required propulsion power, and hence cost of the propulsion plant, could increase due to the need to meet the same speed requirements on a shorter hull). Note, there is an ill-defined limit beyond which further reducing ship size will actually cause it to be more expensive, because the ship's design becomes so cramped its production cost (outfit cost in particular) will rise considerably.

When considering the effect of selection decisions on ship size, the naval architect should keep in mind its reliability (discussed in Section 6.4.4). In addition, the fact that the set of designs is both finite and static (i.e., does not increase in size during the selection process) also is important. Due to these reasons, not every particular combination of interest will be available in the set of designs (Section 3.4.2). This means one can very well take a selection decision that, for the given set, will give an increase in size which is not caused because the design needs to be bigger persé. Instead, it becomes bigger because smaller variants that comply with the selection decision simply were not available in the set, leading to a necessary but artificial growth in size.

The examples also illustrate the ability of the selection approach to identify and remedy flaws. For example, a minor flaw was present in the parametric ship description, namely, the uptakes should have been constrained to a position aft of the radar mast to prevent smoke from interfering with the radar's operation. This flaw was addressed during the selection process.

Decision group no. 1: relation between height of the sensor mast and displacement

First step of the investigation focusses on radar height. By selecting configurations with very high radar positions, the impact on the displacement of the configuration can be investigated. This displacement can then be compared to the displacement of all configurations in the set.

Four sequential decisions were made:

1. Select configurations with the highest radar position. See Fig. 7.13.
2. For the remaining configurations, select those which have the uptakes positioned aft of the radar mast. See Fig. 7.14.
3. For the remaining configurations, select the three smallest designs (length and displacement). See Fig. 7.15.
4. Visualise the smallest configuration. See Fig. 7.20.

It is appropriate to consider how these three decisions affect the displacement of the resulting configurations. To this end, Fig. 7.16 to 7.19 show how the displacement and length are distributed within the set of remaining designs. The displacement changes after decisions no. 1 and 2 in Fig. 7.16 and 7.18. The length changes are shown in Fig. 7.17 and 7.19.

The four images show that, compared to the displacement in the original set, the displacement first increases from 3726 ton to 3888 ton after decision no. 1, and increases further to 3947 ton after decision no. 2. Decision no. 3 does not change the minimum displacement, it just narrows down the set of configurations to the three smallest designs. The overall increase in displacement compared to the original set is 221 tons. Note that the original set also has configurations with lower radar positions (see Fig. 7.13). Something similar happens with increase in minimum length, though after decision no. 1 minimum length increases from 110 m to 115 m and subsequent decisions do not increase minimum length further.

Fig. 7.13 to 7.19 illustrate that a common request in warship design: increase the height of the radar mast, does not come freely, as it increases both ship length and displacement. It thus shows one of the interactions between systems, configuration and performance (Section 1.4).

When interpreting results as in Fig. 7.13 to 7.19, must consider the reliability of the size estimate discussed in Section 6.4.4. Nonetheless, it is likely that increased sensor height causes an increase in displacement. For example, the increase in the vertical centre of gravity a higher radar position brings will necessitate an increase in beam, which increases resistance (among other things) and thus displacement due to the increase in propulsion system weight and additional fuel capacity.

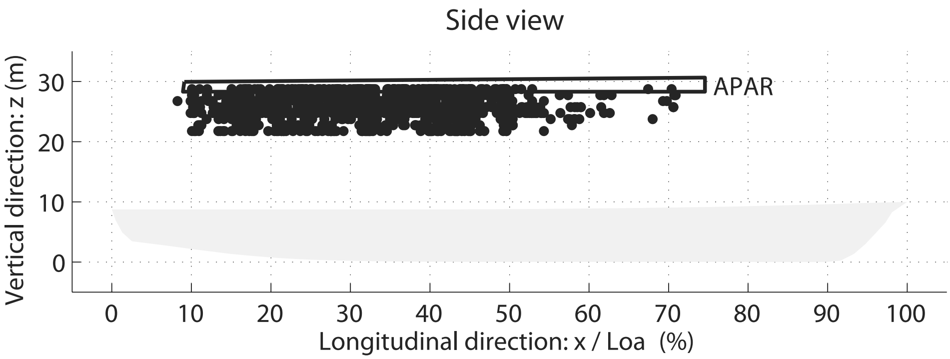


Fig. 7.13 – Decision no. 1: configurations with highest radar position

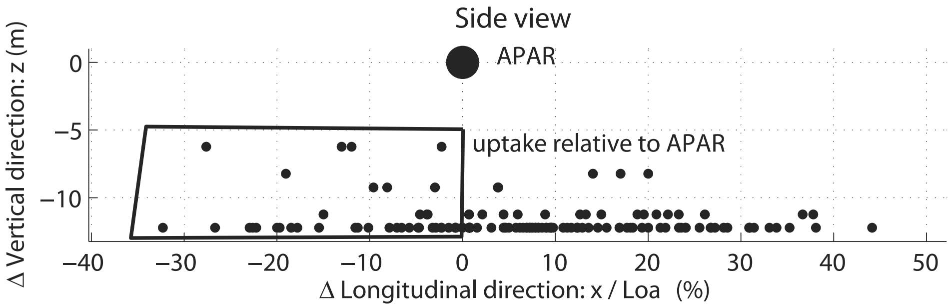


Fig. 7.14 – Decision no. 2: configurations with uptake aft of radar

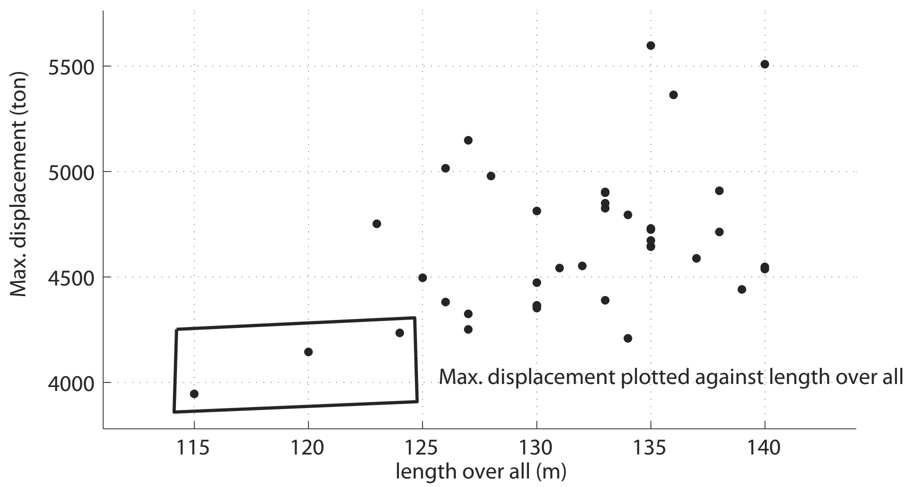


Fig. 7.15 – Decision no. 3: select three smallest configurations

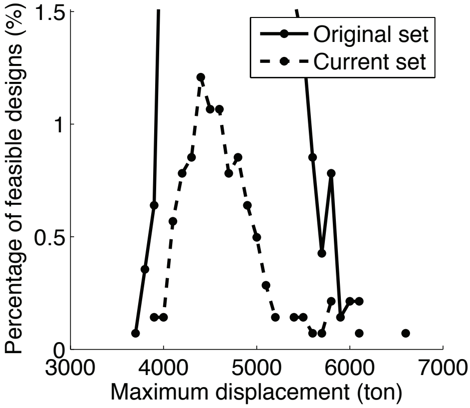


Fig. 7.16 – Distribution of displacement after selection decision 1

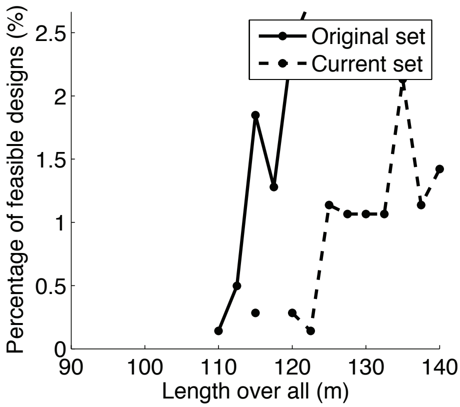


Fig. 7.17 – Distribution of length after selection decision 1

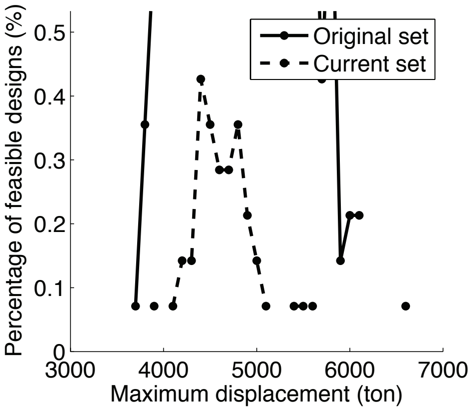


Fig. 7.18 – Distribution of displacement after selection decisions 1 and 2

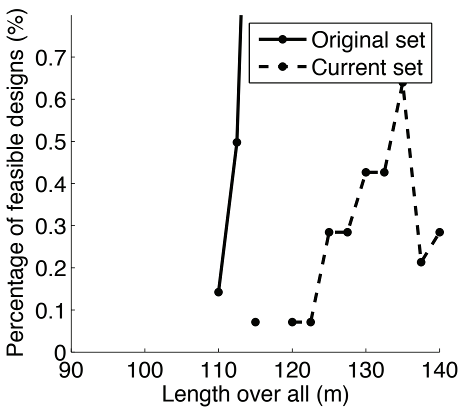


Fig. 7.19 – Distribution of length after selection decisions 1 and 2

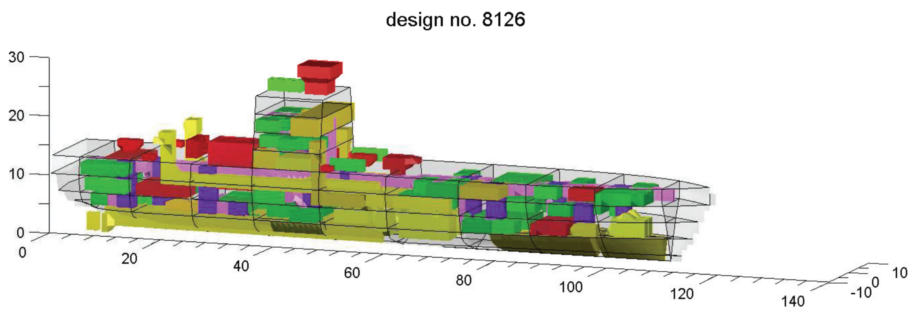


Fig. 7.20 – Configuration no. 8126: smallest acceptable configuration in the set remaining after decisions 1 to 3

Decision group no. 2: concurrent operation of weapon systems

The second decision group looks into the concurrent operation of weapon systems, for which overlap of firing arcs of weapon systems must be considered. More specifically, weapons must be arranged such that those overlaps that prevent concurrent use are removed. Such concurrency requirements imposed on the use of weapon systems can be regarded as an extension of the overlap rules laid out earlier in Section 4.4.3 (see Table 4.1). The impact of these concurrency requirements on both ship displacement and length will be considered. Note that the sequence of decisions should not matter in this instance, as the concurrent operation of specific sets is required for all weapon systems, not just on specific pairs.

Second system → First system ↓	VLS	3-inch gun	CIWS aft	CIWS forward
VLS	-	no	no	no
3-inch gun	no	-	irrelevant	yes
CIWS aft	no	irrelevant	-	irrelevant
CIWS forward	no	yes	irrelevant	-

Table 7.1 – Rules governing overlap between firing arcs

The systems under consideration are listed in Table 7.1, which states whether pairs of firing arcs may overlap. Two items in Table 7.1 warrant further explanation. First, there cannot be overlap between the forward and aft CIWSs, due to the fact they fire in opposite directions. The same happens for the gun and the aft CIWS. Second, overlap between the forward CIWS and the gun cannot be avoided. A position where the CIWS is located aft of the gun was chosen to enable to gun to fire at very close proximity (e.g., the ability to perform 'a shot across the bow'). This means the prevention of overlap focusses on the VLS with three other weapon systems.

This results in five decisions, together with their selection polygons shown in Fig. 7.21 to 7.25.

1. Place 3-inch gun forward of VLS. See Fig. 7.21.
2. Place forward CIWS forward of VLS. See Fig. 7.22.
3. Place aft CIWS aft of VLS. See Fig. 7.23.
4. Place uptakes aft of radar mast. See Fig. 7.24.
5. Select three smallest designs. See Fig. 7.25.

The impact of the preferred arrangement of weapon systems on the ship's length and displacement is moderate. Fig. 7.26 and 7.27 show both length and displacement after decisions 1 to 4. The change in minimum length is from 110 to 114 m, while the displacement increases from 3726 tons to 3920 tons when proceeding from decision no. 1 to no. 4 (a total increase of 194 ton). This increase is considered moderate and rather similar to the increase in the previous decision group. As such, avoiding unwanted overlap of firing arcs does not result in a large increase in ship length or displacement. It should therefore be relatively straightforward to include the concurrent operation of weapon systems in the selection of a promising AAW-frigate.

Of the three remaining configurations, configuration no. 9120 has the smallest displacement. It is shown in Fig. 7.28. One notable feature of this configuration is the location of the VLS directly aft of the superstructure. Normally, this space is taken up by flight deck and hangar. However, their absence means this location in the ship can be used to house weapons, such as the VLS.

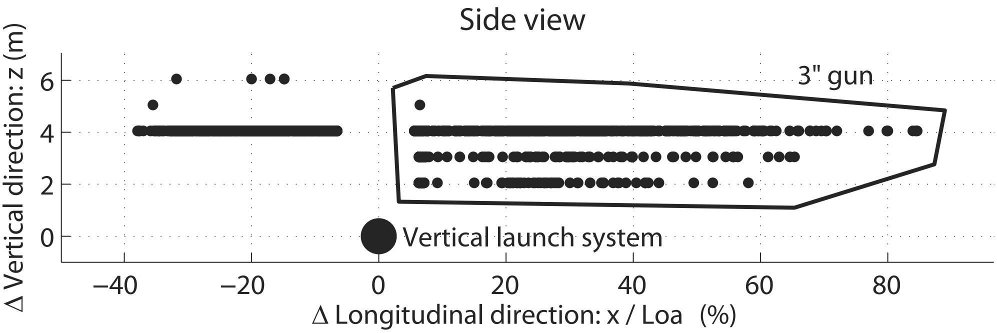


Fig. 7.21 – Decision no. 1

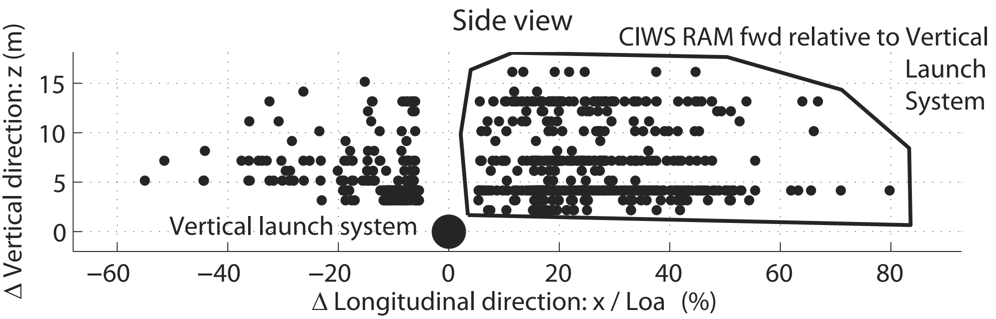


Fig. 7.22 – Decision no. 2

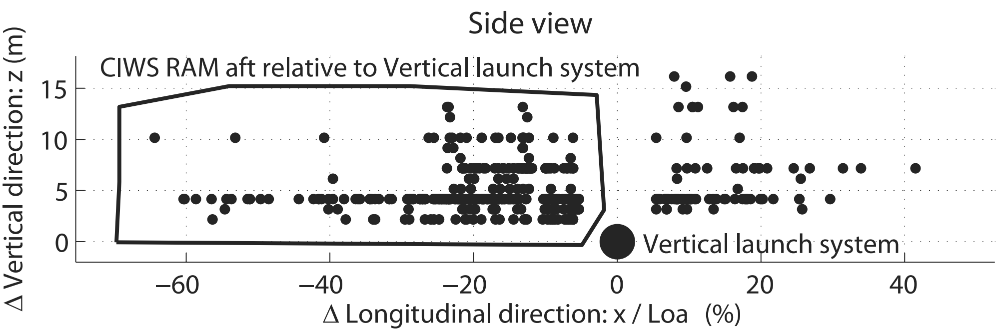


Fig. 7.23 – Decision no. 3

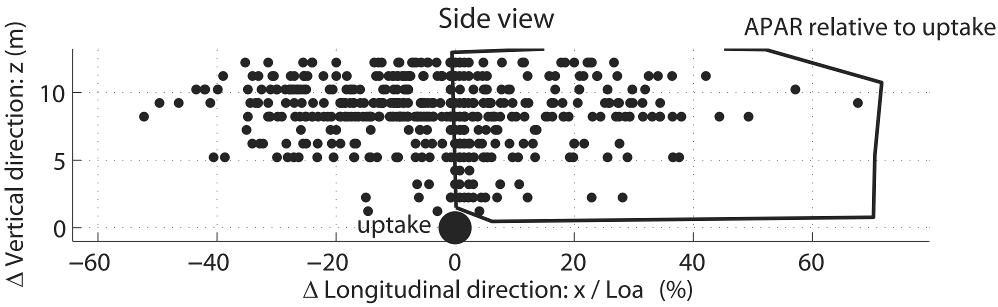


Fig. 7.24 – Decision no. 4

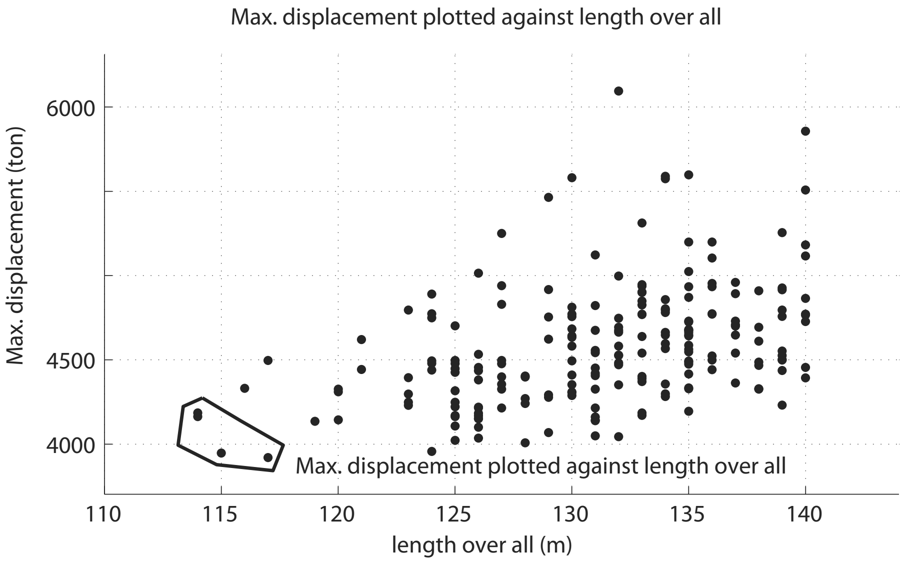


Fig. 7.25 – Decision no. 5

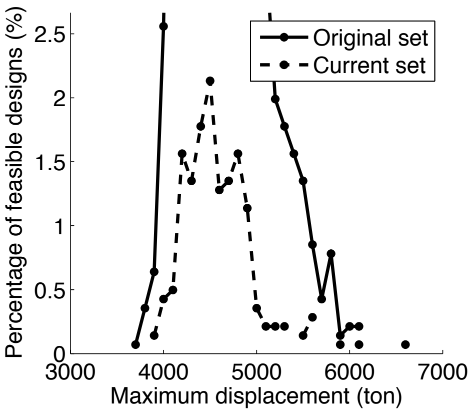


Fig. 7.26 – Distribution of displacement after selection decisions 1 to 4

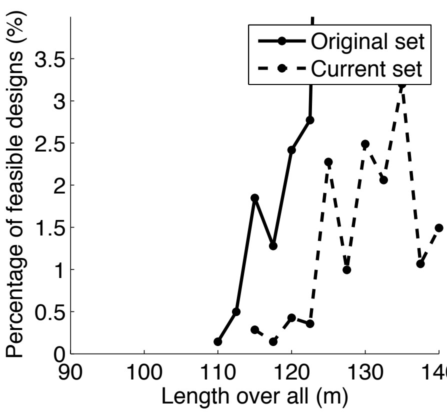


Fig. 7.27 – Distribution of length after selection decisions 1 to 4

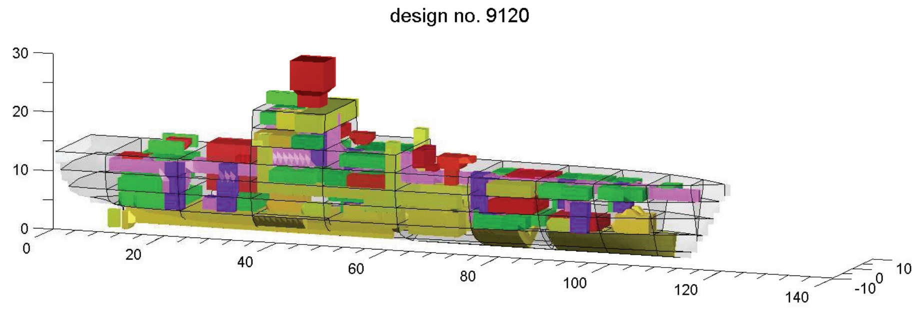


Fig. 7.28 – Configuration no. 9120: smallest displacement remaining after decisions 1 to 4

Decision group no. 3: reduce motion-induced fatigue

Operability of warships in waves can have a considerable impact on the ability to carry out its mission. The AAW-frigate does not have to operate helicopters or small boats. Therefore, crew exposure to motion-induced fatigue and sickness, instead of system operability, poses the biggest risk to reduced operability in higher sea states.

The systems considered from an operability point of view are listed in Table 7.2.

System number	System Name
17	Bridge
54	CIC
58	Ship control center

Table 7.2 – Systems of objects considered to be relevant to operability

Again, several sequential decisions are taken. For brevity, the choices related to positioning the systems in Table 7.2 aft of amidships have been grouped together. The positions were restricted to around 30 % of the length measured from the stern $\pm$ 20%, a common rule-of-thumb used by naval architects to identify positions with limited pitch-induced heave motions (see *Watson [138]*). In addition, by restricting system positions to lower decks, they should become less sensitive to roll-induced accelerations.

1. Select those positions so that the CIC is located between 10 % and 50% of the length over all (measured from the stern), see Fig. 7.29 for the selection polygon.

2. Select those positions so that the bridge is located between 10 % and 60% of the length over all (measured from the stern), see Fig. 7.30 for the selection polygon.

3. Select those positions so that the Ship Control Centre is located between 10 % and 50% of the length over all (measured from the stern), see Fig. 7.31 for the selection polygon.

4. Ensure that the uptakes are located aft of the radar-mast, see Fig. 7.32 for the selection polygon.

5. Select the three smallest designs considered displacement and length, see Fig. 7.33.

One of the three configurations, configuration no. 8126, is shown in Fig. 7.34. Another configuration (no. 4200) is shown in Fig. 7.35. In comparison, this latter configuration is more appealing because closer inspection (using the two-dimensional deck view used earlier, see Fig. 6.29) reveals that most of the crew habitability areas in configuration no. 4200 are located in the aft part of the ship, unlike those in configuration no. 8126.

Intriguingly, configuration no. 8126 in Fig. 7.34 was also found in decision group no. 1 (it was shown earlier in Fig. 7.20). Apparently, the selection decisions in both decision groups no. 1 and no. 3 do not remove configuration no. 8126, because the two decision groups do not impose conflicting demands on it. As such, configuration no. 8126 can comply with both groups of selection decisions simultaneously.

The selection of configuration no. 8126 by two different decision groups is important. It illustrates how two different sets of selection decisions can lead to the same design. Configuration no. 8126 should therefore be more acceptable to the naval architect than those configurations that only comply with one set of selection decisions. After all, there are more reasons (the rationale behind the decisions in both groups no. 1 and no. 3) to like and hence, choose configuration no. 8126, than there are reasons to do so for other configurations.

Most importantly, choosing configuration no. 8126 means that radar height (the performance to be improved by decision group no. 1) and operability in waves (to be improved by decision group no. 3) do not have to be traded off against each other.

Choosing configuration no. 4200 (see Fig. 7.35) over configuration no. 8126, is different in this respect, and must result in a trade-off. Since configuration no. 4200 only complies with decision group no. 3 (and not with decision group no. 1), its selection means more importance has to be attributed to operability in waves than to radar height, i.e., it forms an example of trading off one performance against the other.

In summary, the selection of configuration no. 8126 illustrates the selection approach can help to identify trade-offs by taking different groups of selection decisions and comparing the resulting designs. With this insight, the naval architect can either avoid a trade-off (by choosing configuration no. 8126 in the example above), or settle it using a compromise decision when avoiding a trade-off proves impossible (e.g., by choosing configuration no. 4200 through attributing operability in waves a higher importance than the radar height).

The selection decisions no. 1 to 4 will also impact the range of ship dimensions. Fig. 7.36 to 7.39 show the changes in the range of displacement and length after selection decisions. Between the first and fifth selection decision, the minimum displacement increases from 3726 tons to 3947 tons, while minimum length increases from 110 to 114 m. Fig. 7.36 to 7.39 illustrate that the selection decisions aimed to improve operability do not necessarily lead to a large increase in displacement and length.

The decisions in group 3 illustrate how engineering judgement can be included during the selection process. This is important, because Section 3.5.2 argued that the ability to do so is crucial to address the lack of completeness of ratings and any flaws that may result from it.

Predicting actual sea-keeping performance would obviously enable a far more thorough evaluation of the ship's operability than just resorting to the use of human engineering judgement, but that was not the purpose of the example.

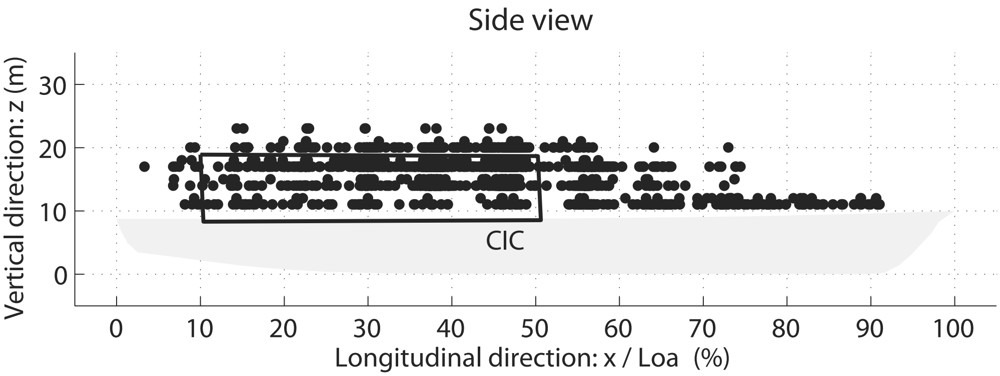


Fig. 7.29 – Decision no. 1

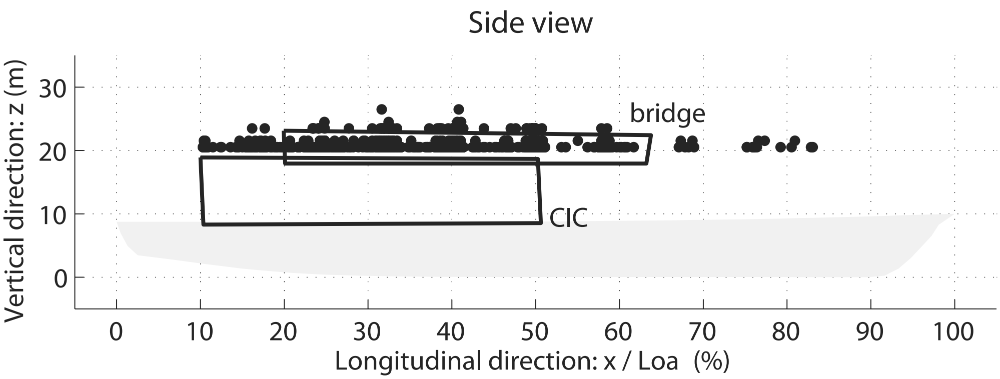


Fig. 7.30 – Decision no. 2

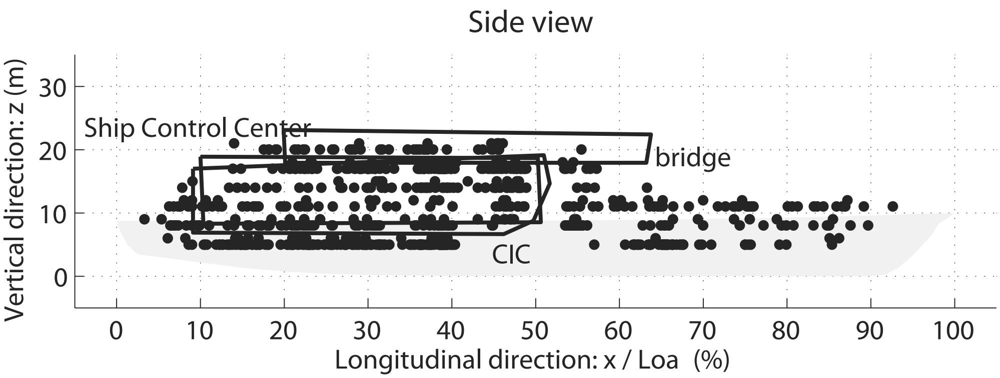


Fig. 7.31 – Decision no. 3

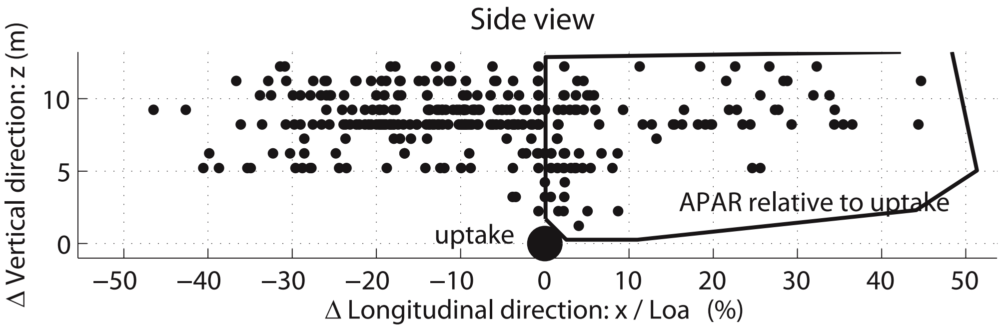


Fig. 7.32 – Decision no. 4

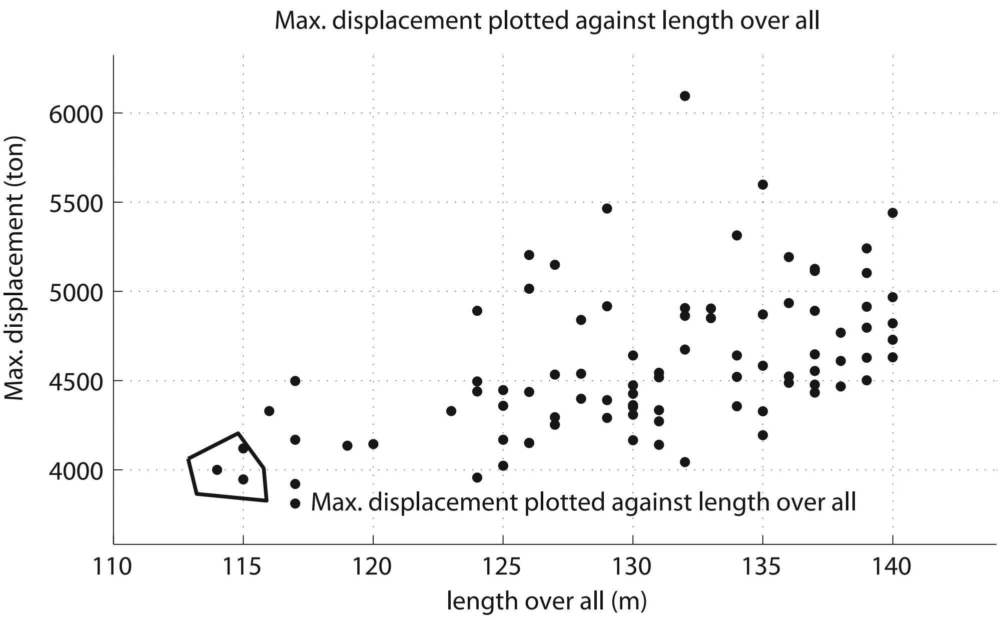


Fig. 7.33 – Decision no. 5

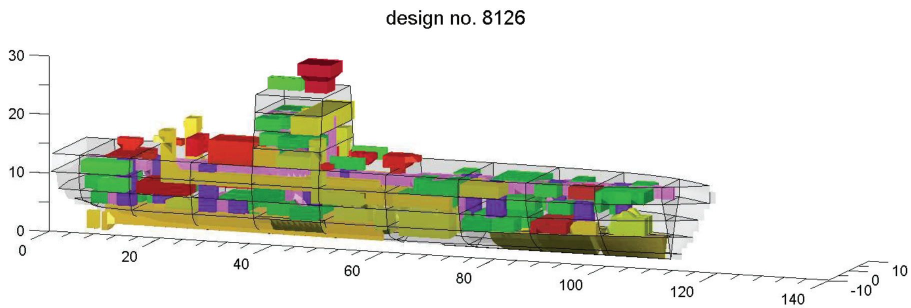


Fig. 7.34 – Configuration no. 8126: small configuration remaining after decisions 1 to 4

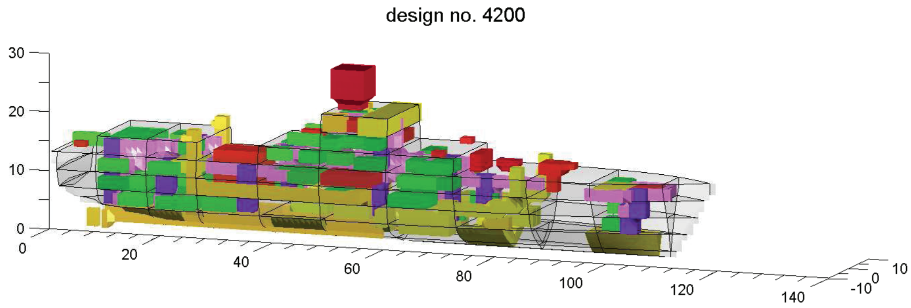


Fig. 7.35 – Configuration no. 4200: small configuration remaining after decisions 1 to 4

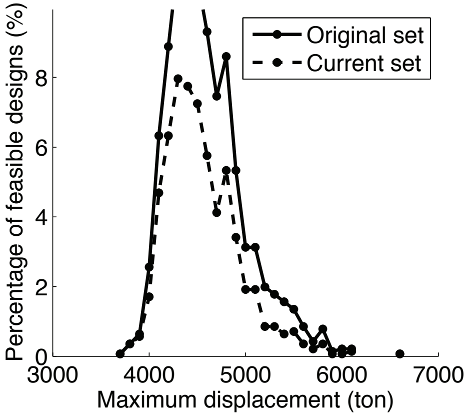


Fig. 7.36 – Range of displacement after decision no. 1

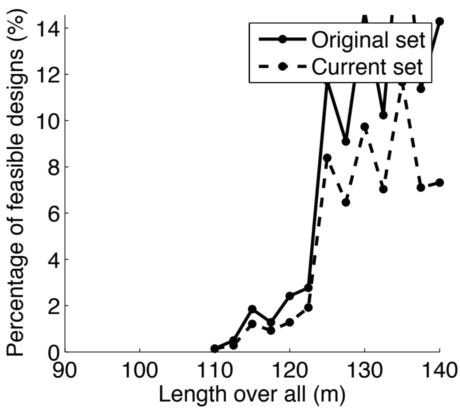


Fig. 7.37 – Range of lengths after decision no. 1

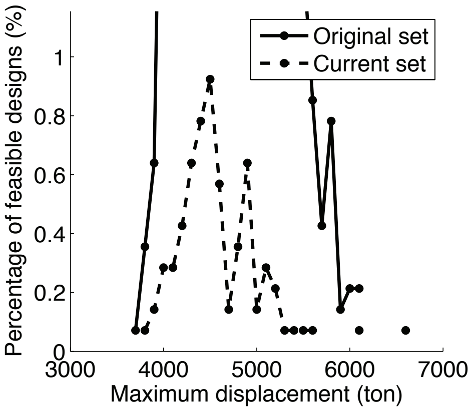


Fig. 7.38 – Range of lengths after decision no. 4

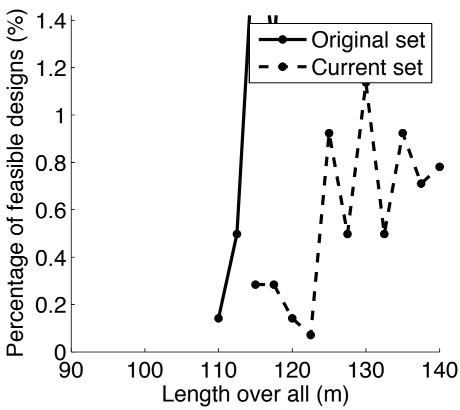


Fig. 7.39 – Range of displacement after decision no. 4

7.5 Discussion of the selection approach

The three decision groups discussed in Section 7.4 offer a glimpse how the naval architect can exploit the selection approach to investigate how different configurations result from different sets of priorities. Examples include selections based on preferring height of the sensor mast over concurrency of weapon system operations. The resulting configurations were shown in Fig. 7.20, 7.28, 7.34 and 7.35.

Some of the selection decisions concerned numerical properties of the configurations, e.g., length over all and displacement. Others considered configurational aspects. Examples of the latter include the position of systems relative to the envelope, e.g., the bridge and CIC, as well as the positions of systems relative to other systems, e.g., uptake and radar. The decision groups also illustrate how flaws in the configurational model can be identified and addressed, e.g., by ensuring that only those configurations are selected that have the uptakes aft of the radar mast. The consideration of operability in higher sea-states using human engineering judgement illustrates how additional knowledge can be included after the search process to resolve the lack of completeness caused by the finite number of performance prediction tools and ratings.

In practice, a more elaborate set of selection decisions would be necessary to investigate and compare alternative configurations more thoroughly. Moreover, the selection approach would be used in conjunction with adjustments to the configurational model and additional search runs to further increase the number of interesting configurations and narrow the focus of the search process (outlined earlier in Fig. 3.6 in Section 3.6). Future applications should help reveal the extent with which this proposed use of the selection approach needs modification.

One last topic warrants attention in the discussion of the selection approach. The selection approach, and in particular its ability to rate and choose designs after the search process (i.e., a-posteriori, see Section 2.4.3), requires the naval architect to invest considerable effort in the investigation of different alternatives. Chapter 3 argued that this is necessary because the formulation of a complete set of rule-based ratings for use by the search algorithm (i.e., a-priori, again see Section 2.4.3) is very difficult at best and would take considerable human effort as well.

This raises the question which of two possible consequences will occur. Firstly, will, on balance, human effort be reduced by the use of the selection approach? Or, secondly, is one activity that requires human effort (formulating rule-based ratings) merely substituted by another (namely, selecting designs), without a net reduction of human effort?

To establish which of the two consequences occurs in practice, it is worthwhile to consider what happens if many rule-based rankings are formulated and then used to guide the search process. Due to the implicit nature of trade-offs (Section 2.4.3), the resulting design with the 'best rating' will, most probably, not reflect the naval architect's wishes. As such, the ratings would need to be adjusted (which takes effort) and the search process repeated (which takes time). This process of adjusting ratings and re-running the search process repeats itself until a design is found that is deemed suitable by the naval architect. Unfortunately, it takes multiple adjustments to arrive at these trade-offs, and hence, a considerable amount of time and human effort. The time spent re-running the search process should not be underestimated, as time available during early stage ship design is generally limited.

With the selection approach, there is no need to define trade-offs prior to the search process, which saves considerable time and effort. Furthermore, it is possible to rapidly test different trade-offs to be used during the selection process without re-running the search process, again saving considerable time and effort, especially since the latter would be needed to adjust the ratings. Therefore, the selection approach is expected to reduce human effort overall, though using it will still take a noticeable amount of time and human effort.

What is also of interest, in relation to effort reduction, is the generation of acceptance for the selected design. The extensive use of the rule-based ratings to settle compromises is still likely to fail to generate acceptance for the resulting designs, even when their use would be successful. Therefore, the ability of the selection approach to generate acceptance is considered reason enough on its own, to use the selection approach over pre-defined rule-based ratings. Without the use of the selection approach, the designs found by the search algorithm would not be used at all.

However, one disadvantage of the selection approach is that designs have to be chosen from a static set, which may cause the selected ship design to be larger than strictly necessary (see Section 7.4). Such an artificial increase in size can be remedied by re-running the search process with additional focussing to create an increased number of interesting designs (see Fig. 3.6 and also Section 8.5).

7.6 Conclusions

This Chapter developed the selection approach that enables the naval architect to identify and select a small collection of promising alternatives from a large and diverse set of ship configurations (based on Fig. 7.1 and repeated in Fig. 7.40).

The selection process provides the naval architect with the ability to thoroughly investigate all relevant characteristics of all feasible designs to gain insight as to why particular configurations should be selected. Filtering supports this investigation by preventing information overload. Filtering takes place by sequentially considering characteristics of interest in the order from high to low importance. Crucially, one can consider and filter both numerical characteristics (using scatter plots) and configurational features, by studying positions of systems in a bespoke visualisation approach (shapes were excluded, though, see Section 7.3.2 and Fig. 7.6). The filterable visualisation of positions helps to trigger the expression of engineering judgement, which is crucial for resolving the lack of completeness. It helps to distinguish between designs with similar ratings, and to identify and remove 'too optimal designs' with flaws resulting from insufficient constraints (adding 'believability', following *Andrews [4]*).

The insight gained in the investigation is used by the naval architect to select designs. Making the naval architect responsible for selection decisions is crucial. It instills a sense of ownership for the results, i.e., a willingness to accept the selected configuration and use it in the remainder of the design process. Selection decisions are expressed using 'hand-drawn' polygons to enclose preferable numerical or configurational characteristics. Selection considers relevant features in a sequential manner from high to low priority. The sequence is frozen during the selection process to work out all relevant consequences of the selection decisions. This helps to resolve the implicit nature of selection decisions and the underlying trade-offs on which they are based.

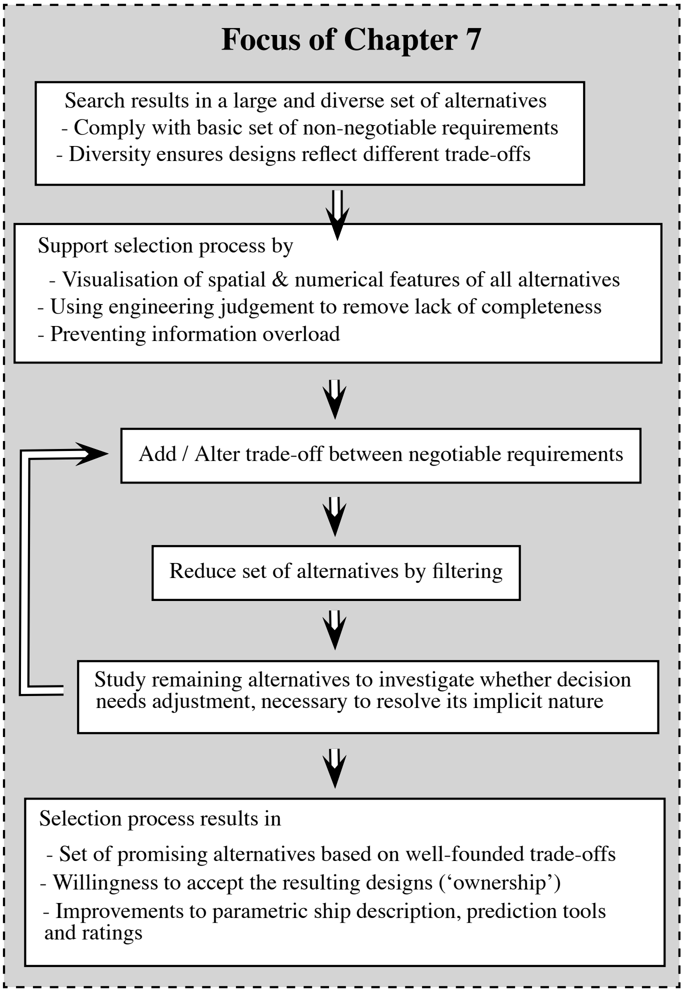


Fig. 7.40 – Focus of this Chapter

Repetitive use of the selection approach enables the naval architect to rapidly investigate different selection decisions and different sequences of priorities to establish whether and how selection decisions must be adjusted in order to arrive at an acceptable design. When compared against the references discussed in Chapter 2 and 4.3, the selection approach presented in this Chapter thus offers naval architects the means to find the proper relative weighings of priorities in an a-posteriori manner (Section 2.4.3) by studying their influence on what matters most, namely, their impact on the ship's design. This differs from the a-priori approaches that define compromises prior to the search process (e.g., as used by *Lee et al. [72]* and *Nick [88]*).

As a closure, the selection approach helps resolve the five issues outlined in Section 3.5.3. The extent with which is summarised below.

- **Provide transparency**. The naval architect can investigate any characteristic of any feasible configuration and use the insight gained to aid the selection of designs. Moreover, using selection, one can investigate interactions between selection decisions in a transparent manner, e.g., study if an upper bound on ship size impacts the range of positions for a drilling derrick. Transparency thus helps the naval architect to learn why a particular set of configurations should be chosen over another set and, crucially, how different selection decisions influence the resulting configuration (see Section 7.4 for examples).

- **Filtering**. The filtering and visualisation of both numerical and configurational aspects allow the naval architect to investigate all relevant characteristics of the feasible designs in a sequential, pair-wise manner, without suffering from information overload (see Figs. 7.3, 7.4 and 7.5 for examples).

- **Address the lack of completeness**. The expression of engineering judgement, triggered by the filterable visualisation of the configurations, helps identify flaws in designs and provides the means to differentiate between designs without formulating additional mathematical ratings, either before or after the search process. Figs. 7.21 to 7.24 show an example. It thus resolves the lack of completeness without relying on extensive human interaction during the search process (in contrast to approaches by *Kim and Cho [67]* and *Buonanno and Mavris [22]*, who both used such human interaction extensively), and without the formulation of a complete set of rule-based ratings prior to the search process.

- **Resolve the implicit nature of trade-off decisions**. This is achieved by allowing the naval architect to rapidly take, consider and revise both the selection decisions and the sequence in which they are made. Thus, a range of different trade-offs can be investigated in a brief time span to assess of the acceptability of the resulting designs. With this insight, the naval architect can establish whether and how the decisions need to be adjusted. In combination, this approach removes the consequences ensuring from their implicit nature.

- **Generate acceptance**. Generating acceptance is achieved by making the naval architect responsible for decision-making. Furthermore, removing flawed configurations helps to improve the feeling that the remaining configurations are indeed 'believable', increasing acceptance.

In summary, the selection approach offers the naval architect the means to investigate the large and diverse set of configurations that should result from the application of the packing-based ship description (Chapter 4) combined with the search algorithm (Chapter 5). Based on the insight gained, and the ability to use engineering judgement to further distinguish between designs and to identify flaws, the naval architect can select different designs reflecting different trade-offs in a rapid manner. The ability to try, consider and revise selection decisions helps the investigation of the impact of different compromises in meeting conflicting requirements, before the few most promising alternatives are selected. As such, the selection approach helps the naval architect to identify and select promising designs from the many possible alternatives that exist (Section 1.3).

The development of the selection approach in Sections 7.2 to 7.3 and its application in Section 7.4 completes the development of the overall design process outlined in Section 3.6 (i.e., Fig. 3.6).

Chapter 8 concludes this thesis by reflecting upon the main findings from Chapters 3 to 7 and studying the extent with which the problems with the early stage design of complex service vessels outlined in Chapter 1 are indeed addressed by the approach presented in Section 3.6 and detailed in the Chapters following that outline.

Chapter 8

Conclusions

8.1 Introduction

This thesis presents an approach for improving the early stage design process of service vessels (Chapter 1). This improvement is necessary to enable the thorough investigation of the many options facing the naval architect, options which concern systems choice, number and topology, the configuration of these systems and the required performance levels. The results of this investigation should help the naval architect to identify promising combinations of systems, arranged in a suitable configuration and with appropriate performances, and thereby help the design of more competitive service vessels.

Chapter 2 started the development of this approach by reviewing relevant efforts from technical literature. Chapter 3 combined those insights gained into a new design approach based on the application of search algorithms. Two perceived benefits justified the use of search algorithms. Firstly, they enable the rapid generation of many alternative ship designs and, secondly, they can ensure these alternatives are feasible, i.e., meet a basic set of non-negotiable requirements. Together, the two benefits should help provide the large number of different ship designs required to investigate the many options, while assuring feasibility allows the naval architect to focus on handling trade-offs between conflicting options or requirements.

Unfortunately, the practical application of search algorithms in early stage ship design is not straightforward. Chapter 3 discussed numerous issues encountered when trying to do so. It also proposed remedies for each of them to ensure that search algorithms can successfully generate the many alternative ship designs needed to investigate the wide range of options (these options concern different systems, different configurations and different performance levels). The issues resolved in Chapter 3 concern the parametric ship description used to describe and alter the ship's design, the objectives and constraints used to guide the search process, and the approach used to investigate and select designs after completing the search process. These remedies were integrated in a new design approach outlined in Section 3.6. This approach was subsequently developed in full in Chapters 4, 5 and 7.

Chapter 4 developed a packing-based parametric ship description suitable for early stage ship design. Most importantly, the description can instigate both large and concurrent changes to the entire ship design by solely using changes in parameter values (and, hence, without relying on human interaction). As important, it is reusable and can describe a wide range of different service vessels. Chapter 5 discussed the search algorithm (NSGA-II), introduced the prediction tools, and developed the constraints and objectives that will guide the search effort towards designs that are feasible and have a high packing density. In combination, the parametric ship description and the search algorithm are able to generate a large and diverse set of feasible configurations, as was shown by two applications in Chapter 6. Chapter 7 completes the development of the process outlined in Section 3.6 by developing the novel approach to support the naval architect in the selection of a small collection of promising alternatives.

This Chapter summarises the main findings of the previous Chapters in Section 8.2 and uses them in Section 8.3 to assess whether the approach does indeed reduce human effort during the early stage design of service vessels; it thus establishes whether the approach achieves the research objective outlined in Chapter 1. Section 8.4 establishes novel features of the approach, while Section 8.5 discusses topics for further development. Section 8.6 concludes this thesis by presenting issues related to its practical application.

8.2 Main findings

This Section summarises the main capabilities of the approach developed to support the early stage design process of service vessels. The problems outlined earlier in Chapters 1, 2 and 3 serve as a starting point and are summarised below.

8.2.1 The practical problem

Chapter 1 posed the practical problem of designing a service vessel as 'finding a ship design with a coherent configuration of relevant systems with suitable performances' that also complies with all non-negotiable design requirements and that forms a suitable compromise between negotiable design requirements.

It also identified several problems naval architects face during the search for such a design:

- First, assessing whether a particular design complies with non-negotiable requirements, and whether it presents a suitable set of trade-offs among negotiable requirements, requires the availability of a ship configuration at a suitable level of detail to predict the required performances, which takes human effort.
- Secondly, there are many options with respect to different systems, system positions in the ship and required performance levels. Investigating these options takes considerable effort, as systems must be selected, a configuration created and performances predicted for a large number of potentially relevant combinations of options.
- Thirdly, design requirements may change due to insight gained from previous design iterations (Section 1.4). Unfortunately, changing requirements and design concurrently further increases human effort, as their effect on the ship's design can be so large that the repetitive adjustment and refinement of a single design often is impossible.

Together, the three problems result in a significant increase in the effort required to investigate to the many options available, which hampers finding a suitable and competitive alternative during the early stage design of service vessels. Hence, **reducing this effort to manageable proportions** to enable the investigation of many alternative ship configurations became the objective of the research presented in this thesis (Section 1.7).

8.2.2 The proposed solution: use search algorithms to generate 3D ship configurations that meet non-negotiable requirements

Chapter 2 reviewed prior computer-support efforts intended for early stage ship design. It identified the use of a search algorithm combined with a parametric ship description integrated with a set of prediction tools as a possible avenue to increase the number of design iterations without increasing human effort, which would allow a larger number of alternatives to be considered in a given time-frame (Section 3.2 and 3.6). An additional benefit is that search algorithms can search for designs that meet non-negotiable design requirements (Section 3.5.2 and 3.6).

However, the application of search algorithms to support the early stage design of service vessels is not straightforward, as is discussed in Section 8.2.3.

8.2.3 Realising the solution: resolving issues that hamper the use of search algorithms during early stage ship design

Section 3.4 discussed the issues to be addressed in order to enable the practical application of search algorithms in early stage ship design. The issues are related to three stages in the application of search algorithms. Firstly, the preparation for the search process, secondly, performing the search and, thirdly, using the results after the search process has finished.

Resolution of these issues is crucial to enable the application of a search algorithm to resolve the problems faced during early stage design of service vessels. Each of the three groups of issues and their remedies is discussed below.

Resolving issues encountered prior to carrying out the search process

Section 3.4.1 and 3.5.1 discussed the issues one encounters when preparing for the search process, as well as their remedies. These issues and their remedies are summarised below.

- **Achieve large and concurrent changes to the entire ship configuration using input parameters**. The parametric ship description must handle large changes to the entire configuration to cope with the large changes in system choices and performance requirements one encounters during early stage ship design (Section 3.5.1). Most importantly, it must do so while solely using changes in input parameter values, and without relying on human inventions during the search process, something Chapter 2 identified to be missing in existing early stage ship design approaches.

  The packing approach developed in Chapter 4 is seen to fill this gap. It uses six object types: ‘envelope’, ‘subdivision’, ‘hard’, ‘soft’, ‘free space’ and ‘connection’ to represent ship systems (Fig. 4.21 to 4.26). A seventh object type: a ‘logical’ object, helps focus the ship description (see Section 4.6). The initial shape and size of the objects results from both static human input (which does not change during the search process) and parameter values from the search algorithm (which do change). The set of objects is packed inside a box-shaped positioning space (Fig. 4.31). Overlap rules determine which object types can overlap (Table 4.1).

Overlap management changes the initial shape and initial position of objects to ensure compliance with overlap rules. Overlap management is applied to sets of objects in a user-defined sequence, until either all objects are packed successfully, or until compliance with overlap rules cannot be achieved, at which point the packing process stops. A constraint counts the number of objects that could not be packed and guides the search process towards configurations that can accommodate all objects so that the objects comply with all overlap rules. An example in Section 4.8 illustrated how changing input parameter values changes the initial shapes and positions of objects. In turn, this changes the final shape and position of objects, which creates different configurations as a result (see Fig. 4.90).

Subsequently, the applications in Chapter 6 illustrated how a single packing-based parametric ship description can generate configurations with different hull and superstructure shapes and sizes, different subdivision configurations and, crucially, different positions and shapes for objects representing systems located inside and on the envelope (see Figs. 6.16 to 6.25, for instance).

As such, the packing approach is able to achieve large and concurrent changes to the entire ship configuration solely by using input parameters and without relying on human interaction.

- **Invention and reusability**. Humans are responsible for inventing the parametric ship description (see Sections 2.4.2 and 3.5.1). It should therefore be reusable, i.e., be able to describe and alter different ship types, in order to recoup the investment of the human effort necessary to create it.

  The two applications in Chapter 6 showed the packing approach from Chapter 4 and, specifically, its object types, overlap rules and design changes for overlap management (Section 4.4.5), can indeed be applied to two different types of service vessels. This was demonstrated by generating designs for a frigate and a deep water drillship, respectively.

Resolving issues encountered during the search process

Section 3.4.2 and 3.5.2 discussed the issues encountered during the search process. These issues, and their remedies, are summarised below:

- **Sufficient yield**. The search algorithm must find a sufficiently large set of relevant alternatives, i.e., it has to have a sufficient yield. Several measures were taken to improve yield. For example, limiting the number of parameters reduces the complexity of the search process. This can be achieved, for instance, by constraining a system to the centre-line of the ship. Similarly, clustering more systems in fewer, larger objects helps to reduce the number of input parameters (Sections 3.5.2 and 4.4.1). Also, the use of focussing (Section 4.6) helps steer the search process towards configurations the naval architect deems to be promising. All these measures should help to increase yield, which will give the naval architect more designs to choose from.

Furthermore, the finite number of designs considered by the search algorithm means the search process is unlikely to find a 'perfect' design (Section 3.4.2). Therefore, dedicated changes will remain necessary after the search process, even for promising designs. These can be investigated using existing ship design approaches, as is covered in Chapter 2.

The two applications in Chapter 6 differed with respect to their yield, but both were able to generate several hundreds to several thousands of designs, all of which complied with the basic set of non-negotiable design requirements for which constraints were formulated. A set of feasible designs this large is deemed to be sufficiently big to allow the selection approach developed in Chapter 7 to investigate a wide range of trade-offs.

- **Lack of completeness of ratings**. Search algorithms need objectives and constraints to guide the search process. However, their number and, hence, their ability to distinguish between designs, is finite (Section 3.4.2). One must handle this 'lack of completeness' of ratings to ensure the search algorithm finds relevant designs, while simultaneously preventing the generation of erroneous designs that may result from insufficient constraining.

  The lack of completeness is addressed by using a combination of rule-based constraints and objectives, together with human judgement.

  – **Rule-based constraints and objectives**. A small set of important constraints are formulated as mathematical rules to guide the search process towards configurations that comply with a basic set of non-negotiable design requirements, such as the ability to float upright in stable condition whilst meeting the speed and endurance requirements (Section 5.4.2).

    A single objective is used, namely, packing density, to search for the densest designs (Section 5.4.3). No other rule-based objectives were formulated to avoid spending effort on inventing objectives that settle trade-offs. The implicit nature of trade-offs caused the author to consider it not worthwhile to formulate such rule-based objectives prior to the search process (Section 3.4.2).

  – **Human judgement**. Human judgement is used to rate designs after the search process to resolve the lack of completeness (Section 3.5.3). This avoids the need to use human interaction during the search process, which would slow it down, and it enables designs to be rated on features that lack rule-based ratings.

    Since one does not know beforehand what additional knowledge will be expressed, it is important that the set of designs is diverse (Section 3.5.2). This diversity helps the investigation of different trade-offs as is revealed by reversing an important observation. If different trade-offs result in different designs then different designs must reflect different trade-offs, even if those trade-offs are unknown to the naval architect.

    Diversity is created by exploiting the randomness introduced by selection, crossover and mutation operations used by NSGA-II during the search process (Section 5.2.3). Use of a dedicated objective to increase diversity was considered and rejected in Section 5.4.1.

    Human judgement also offers the means to address any flaws that result from insufficient constraining ('too optimal' designs, Section 3.4.2). For example, Section 7.4 showed how a constraint could be added after completing the search process which ensured exhaust gases did not interfere with radar usage. The expression of human judgement is explained in more detail below.

Chapter 7 showed how the combination of the packing and selection approaches successfully address the lack of completeness; one example used rules-of-thumb to select designs with better operability in a seaway (see Fig. 7.29 to 7.35).

Resolving issues encountered after the search process

Finally, Section 3.4.3 and 3.5.3 discussed the issues encountered after the search process completed. These issues, and their remedies, are summarised below.

- **Transparency**. The naval architect must be able to investigate all characteristics of interest: numerical and spatial, of all designs in the set of feasible configurations. This helps the naval architect to learn why particular alternatives are preferable over others (Section 3.5.3). Transparency is enhanced further by the selection approach; one can rapidly investigate how different selection decisions result in different designs. The examples in Chapter 7 illustrate how AAW-frigate configurations can be transparently selected so that they excel at particular performances.

- **Information overload**. Investigating all characteristics of interest, of all designs, risks creating an information overload. This was resolved by providing filtering of both numerical and spatial features. By considering features pairwise, in a sequential manner from higher to lower priority, one can still consider all relevant features of all designs without loosing insight (Section 7.3.2). Considerable attention was paid to develop a filterable visualisation that can show the positions of objects for many designs concurrently (Section 7.3.2). The examples in Chapter 7 illustrate how filtering successfully enables the naval architect to consider the relevant features of all feasible designs without becoming swamped in information.

- **Lack of completeness**. The previous discussion already mentioned the use of engineering judgement to address the lack of completeness by having a naval architect rate designs after the search process. This is facilitated by visualising object positions in a suitable context, namely, via a top-view and a side-view general arrangement (Fig. 7.4 and 7.5). This provides a context familiar enough for the naval architect in which to express relevant experience.

  Using human experience requires considering all feasible designs generated in the search process, not just those deemed ‘optimal’. The reason is simple, adding knowledge may make ‘sub-optimal’ designs ‘optimal’ (Section 3.5.2). However, since one does not know beforehand which designs will become ‘optimal’, one must study all feasible designs to identify those ‘newly optimal’ designs. This does significantly increase the number of designs to be studied and this then further emphasises the need for filtering.

  As was shown in Chapter 7, the selection approach can address the lack of completeness, e.g., for operability in a seaway (see Fig. 7.7 and Figs. 7.29 to 7.35).

- **Implicitness of trade-offs between negotiable requirements**. One reason to limit the use of pre-defined rule-based objectives was to avoid spending effort on formulating rules that settle trade-offs between negotiable requirements in an a-priori manner, as these rules depend on trade-offs that are implicit and that may change depending on the resultant designs (Section 2.4.3).

The implicit nature of trade-off decisions is removed by increasing the speed with which the consequences of trade-off decisions will become known. This is achieved in two ways. Firstly, ensuring compliance with a basic set of non-negotiable requirements means the naval architect is relieved from ensuring the designs are feasible and thus can focus on investigating trade-offs. Secondly, the selection process in general (discussed in Chapter 7), and the use of filtering based on 'hand-drawn' polygons in particular, help to investigate the consequences of different trade-offs in a rapid manner. This is done by quickly removing designs from the entire set until only those designs remain that comply with all selection decisions. The resulting designs can then be studied, which gives one the insight one needs to reflect upon the selection decisions and, if necessary, to adjust them.

The examples in Chapter 7 illustrated the simplicity and the speed with which different trade-off decisions can be investigated, as well as the ability to quickly reveal the different designs that result from these decisions.

- **Acceptance**. Without a sense of ownership, it is considered that the results from the search process will not be used, as it is likely that nobody would feel responsible for the resulting designs. Generating the willingness to accept designs is therefore seen to be crucial and is achieved in three complementary ways. First and foremost, the naval architect is responsible for all selection decisions (Section 3.5.3). Secondly, the use of human judgement offers the means to remove flawed (and hence unacceptable) designs that occur due to insufficient constraining, i.e., those designs that are 'too optimal'. Lastly, investigating all feasible designs (with good and bad performance levels) provides the naval architect with insight why a particular design is to be preferred over others.

8.2.4 The approach implementing the solutions

The main findings from Section 8.2 were implemented in the overall process shown in Fig. 3.6 and reproduced in Fig. 8.1. The steps in Fig. 8.1 are summarised below with reference to the relevant Sections of Chapters 4 to 7 .

1. A **database with human inventions** stores the individual parts of the parametric ship description, the performance prediction tools, the objectives and constraints, and the search algorithm. All were invented prior to the search process and were discussed, respectively, in Section 4.4, 6.2.2 and 6.3.2 (ship parts), Section 5.3 (performance prediction) and Section 5.4 (ratings) and Section 5.2 (search algorithm). The database facilitates reusability of these parts, which was shown by the two applications in Chapter 6.

   Both applications used the same packing approach (i.e., the same object types, overlap rules and design changes for overlap management), but obviously had to use the same object types to represent different systems in the parametric ship description. For example, the frigate description does not have a drilling derrick defined by a hard object, nor does the drillship description have a hangar defined by a hard object. Also, both applications shared most performance prediction tools and constraints, and also used the same search algorithm: NSGA-II.

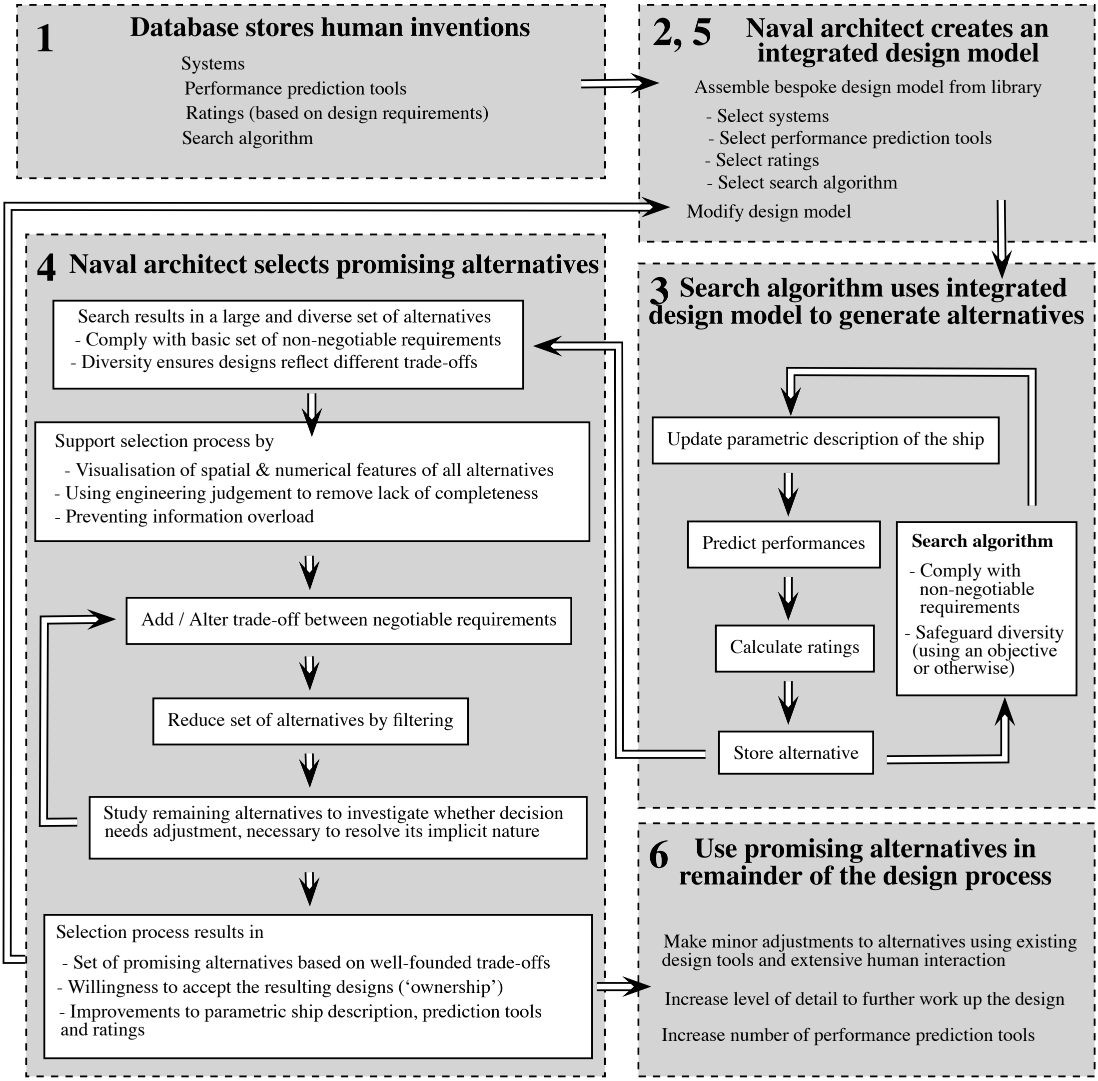


Fig. 8.1 – Outline of the approach for the early stage design of service vessels (reproduced from Fig. 3.6)

2. **Create an integrated design model** is the next step in the process. The naval architect assembles the relevant systems of objects into a bespoke parametric ship description that the search algorithm can adjust by changing its parameter values (Chapter 4 and 6). Subsequently, the naval architect integrates the parametric ship description with numerous prediction tools (Section 5.3) linked to objectives and constraints (Section 5.4) that are used by the search algorithm (Section 5.2). The resulting integrated design model will allow the search algorithm to generate the many alternative ship designs in the subsequent step (as shown by the applications in Chapter 6).

3. **Generate alternatives**. Once assembled, the parametric model, performance prediction tools and ratings are used by the search algorithm to generate a large and diverse set

of alternatives (Section 3.5.2) that comply with the non-negotiable design requirements included as constraints (Section 3.5.2). The diversity needed to cover a wide range of trade-offs (Section 3.5.2) is maintained by exploiting the randomness introduced by the genetic operations used by NSGA-II (Section 5.4.1). Chapter 6 illustrated the ability to generate a large and diverse set of alternatives (see Tables 6.1 and 6.8) by exploiting the ability of the packing-based ship description to handle large and concurrent changes to the entire ship configuration.

4. **Select promising alternatives**. The naval architect uses the selection approach from Chapter 7 to study the large and diverse set of feasible alternatives in a transparent manner. It helps the naval architect to gain insight into trade-offs between negotiable design requirements and to learn how the different options with respect to systems, configuration and performances interact and influence the overall design (Section 3.5.3).

   Extensive use of both filtering and visualisation helps to create this insight. Most importantly, visualisation of object positions provided in a proper context, namely, a general arrangement plan, triggers the crucial expression of human engineering judgement to distinguish between designs on aspects for which no rule-based ratings have been formulated. This helps to resolve the 'lack of completeness of ratings', which simultaneously helps to identify 'too optimal' designs (Section 3.5.3).

   Once sufficient insight has been gained, the naval architect can select a small set of promising alternatives that form a suitable compromise between negotiable requirements. Most importantly, the responsibility for selection decisions resides solely with the naval architect to generate the critical acceptance of preferred designs as the outcome of the selection process (Section 3.5.3).

   Chapter 7 illustrated how the selection approach can successfully help the naval architect to select different designs reflecting different trade-offs in a transparent manner (from the large set of feasible alternatives produced in the manner shown in Chapter 6), with full acceptance for the results. It also showed how the approach fosters the expression of engineering judgement, which is crucial to help identify flaws and to distinguish between designs.

5. **Alter the integrated design model.** The investigation of the feasible alternatives may reveal the need to alter the parametric ship description, the performance prediction tools and the ratings used in the search process. Humans can make such alterations and study their impact by repeating steps 2 to 4 above. The applications in Chapter 6 illustrated this ability by changing the weapon and sensors fit in the frigate application to create variants specialised in air defence and anti-submarine warfare, respectively.

6. **Refine promising alternatives.** The investigation and selection process can lead to a sufficiently interesting and acceptable design that should be considered in more detail. Such a more detailed investigation can be performed using existing early stage design approaches from Chapter 2, e.g., *Andrews and Pawling [8]* or *Van Oers et al. [131]*. Note, such refinement was not pursued in the research presented in this dissertation.

8.3 Conclusions: does the approach reduce human effort?

Ultimately, the approach presented in this dissertation must reduce the effort required to investigate many alternatives during the early stage design of service vessels in order to meet the research objective stated in Chapter 1.

This Section considers whether the research objective is actually met by reviewing claims made in Section 3.7. These claims listed the ways in which the approach outlined in Section 3.6 and developed in Chapters 4, 5 and 7 should be able to reduce effort. They are reviewed below.

- **Claim 1:** *'The human effort required to invent parts of the ship's description cannot be reduced. It can be made worthwhile, however, by ensuring reusability over several design projects. Reusability is enhanced further by using a suitable break-down that supports customisation for different types of service vessels'.*

  Review 1: Considerable effort was indeed spent developing the ship description, i.e., the packing approach in Chapter 4. Once developed, though, the object types, overlap rules and design changes for overlap management were reusable for two different ship types, as illustrated by the applications in Chapter 6. The reusability makes the investment worthwhile.

- **Claim 2:** *'The application of a parametric ship description that relies solely on input parameters to achieve large changes to the entire ship description reduces human effort by removing the need for inventions normally required to handle these changes during the generation of alternatives'.*

  Review 2: The simple example in Section 4.8 and the elaborate applications in Chapter 6 illustrate that large changes to the entire ship description can be instigated successfully by relying solely on changes in parameter values, instead of by relying on human interaction. Hence, the effort reduction is realised.

- **Claim 3:** *'The choice to use search algorithms to search for alternatives that meet a basic set of non-negotiable requirements formulated as constraints relieves the naval architect of the burden of ensuring compliance with these requirements. It reduces human effort considerably as, in practice, the focus tends to be firstly meeting non-negotiable requirements and then considering negotiable ones'.*

  Review 3: The applications in Chapter 6 illustrate the search algorithm's ability to find a set of configurations that comply with the non-negotiable requirements on draught and meta-centric height (Figs. 6.13 and 6.15). As such, the naval architect is relieved from safeguarding the non-negotiable requirements formulated as constraints, which means the effort reduction is realised.

- **Claim 4:** *'The choice to use search algorithms to search for different alternatives provides the naval architect with a range of designs that reflect different compromises. It thus removes the need to formulate ratings to cover each and every compromise one might encounter'.*

  Review 4: The applications in Chapter 6 show the approach is able to generate a large and diverse set of alternatives. Examples of the application of the selection approach from Chapter 7, subsequently revealed this diversity can be successfully used to investigate different trade-offs for which no ratings were formulated by using human engineering judgement, e.g., to study the effect of improved operability in waves on the overall ship configuration. It thus reduces human effort by removing the need to formulate a large number of rule-based objectives and, instead, enabled the use of a single, simple objective, namely, packing density (see Section 5.4).

- **Claim 5:** *'The availability of a set of alternatives with different system choices, configurations and performance levels that meet non-negotiable requirements helps the naval architect to get rapid feedback on different compromises in a transparent manner, as working out the consequences of the trade-off can be performed in a simple and rapid manner by filtering the set of alternatives, before settling on the final decision. It thus reduces the effort required to address the implicit nature of compromise decisions'*.

  Review 5: Removing the implicit nature of trade-off decisions can be achieved only by giving insight in all relevant consequences of the decisions. With this insight, the naval architect can consider whether the trade-off decisions need adjustment.

  The speed with which this insight is gained is crucial. A long delay between taking a decision and gaining the insight in its consequences hampers the investigation of numerous decisions and, in turn, makes it difficult to adjust trade-off decisions such that all their consequences become acceptable.

  Fortunately, both the speed of decision-making and the speed of gaining insight are increased by the selection approach developed in Chapter 7 by using three complementary improvements.

  – First and most important, the selection decisions that settle trade-offs are used in an a-posteriori manner, i.e., are used to choose from the large and diverse set of feasible designs generated during the search process. This replaces the more traditional a-priori approach often used in ship design, where decisions are taken, and then used as a basis to create a design. Hence, the benefit of the a-posteriori decision-making is that it removes the time-consuming generation of designs after taking a decision.
  – Second, expressing the selection decisions is simplified to its essence: by drawing a polygon that encloses preferable characteristics of feasible designs. As such, it is very fast.
  – Third, because selection decisions are applied in an a-posteriori manner, the removal of non-compliant designs from the set of designs through filtering is very fast as it does not involve elaborate calculations; a simple check of which designs fall outside the selection polygon suffices.

  By using the three improvements in combination, it becomes possible both to rapidly take trade-off decisions to select designs, and to study the set of designs that comply with these decisions. From this, the naval architect gains insight in whether the decisions need adjustment, or whether they can be retained. Due to the speed of the selection approach, numerous different trade-off decisions can easily be investigated, considered and adjusted.

  As such, the selection approach reduces the effort required to investigate the effect of different compromise decisions in a transparent and rapid manner. Therefore, it reduces the effort to resolve the implicit nature of compromise decisions.

- **Claim 6:** *'The use of visualisation during the investigation of alternatives helps the naval architect to identify 'too optimal' designs. It reduces the effort as one does not have to formulate all ratings required to prevent such flaws. Moreover, the visualisation provides the correct context in which one can distinguish between designs, further reducing the effort required to formulate rule-based ratings'.*

  Review 6: The application of the selection approach in Chapter 7 illustrates the ability the use of human judgement expressed in a proper context to identify flaws (such as the erroneous positioning of uptakes relative to a radar mast) and to increase the ability to distinguish between designs (e.g., to identify designs with improved operability in waves). Human judgement removes the 'lack of completeness of ratings' and its consequences. As such, it allows a considerable reduction of the number of rule-based ratings and, hence, a reduction of the human effort required to invent or adopt them.

- **Claim 7:** *'The human responsibility for taking trade-off decisions helps to generate acceptance for the results. As such, it does not reduce human effort directly; rather, it increases it as one must use the selection approach to achieve this. Doing so, however, ensures the practical acceptance of the alternatives generated by the search algorithms. On balance, the extra investment of human effort will be compensated by the speed with which the search algorithm can generate a large and diverse set of alternatives'.*

  Review 7: Taking selection decisions to resolve trade-offs (such as in the examples in Chapter 7) does take time. However, the availability of a large and diverse set of configurations with predicted performances ensures the decision-making process is quite fast, for two reasons. Firstly, no additional predictions need to be made, which saves time and, secondly, decisions are propagated by filtering the set of designs, which is very fast.

  As a result, generating acceptance by giving the naval architect responsibility for decision-making does increase effort, but this increase is considered to be moderate. Most importantly, this increase is necessary to instill the crucial acceptance of the results. Moreover, the reduction in effort by Claims 5 and 6 means the naval architect has more time available to investigate and consider trade-offs before selecting those alternatives deemed most promising.

Thus, the approach presented in this thesis does reduce effort. It carefully balances moderate increases in effort prior to the search process (necessitated by the naval architect's role in invention) and after the search process (necessitated by the naval architect's responsibility to select designs), with far larger reductions in effort during the search process. These latter reductions are possible due to the use of the packing-based ship description when combined with the NSGA-II search algorithm to find a large and diverse set of configurations that meet basic non-negotiable design requirements.

Most importantly, the effort reduction means the approach presented in this thesis is able to meet the research objective stated in Section 1.7: '**Reduce the effort required to generate and investigate a large number of alternative ship designs during the early stage design of service vessels**'. As a result of this achievement, the number of feasible designs that a naval architect can consider during the early stage design of service vessels should increase considerably. This increase is necessary to help the naval architect make a thorough investigation of the many options of interest (Section 1.3). The insight gained in this investigation should help the naval architect to design more competitive service vessels.

On a more practical note, the applications from Chapter 6 were carried out in a period of three months each (gather data, develop the parametric model and run the test cases). This means that, without considering their quality, it would be necessary to generate, respectively, 75 feasible frigate designs and 214 feasible drillship designs per day, 24 hours a day, for seven days a week, for three months non-stop, to manually come up with the number of feasible frigate and drillship configurations found in the applications in Chapter 6.

8.4 Novel features

The following features of the approach developed in this thesis are considered novel and should contribute to furthering the state-of-the-art in early stage ship design.

1. An investigation of the interaction between the configuration of systems, and envelope shape and size (see Section 4.5.4) that covered both 'packing' (e.g., as in *Van der Nat [120]* and *Nick [88]*) and 'wrapping' (*Andrews [3]*) to establish its impact on the resulting designs.

2. The development, implementation and application of a reusable, packing-based parametric ship description that is able to define and change the entire ship description at a variable level of detail and which is suited for application with search algorithms (Chapter 4). Most importantly, the description is able to handle large changes to shape and size of both hull and superstructure concurrently with changes to the shape and positions of systems inside and on the envelope, unlike any existing approach known to the author.

3. The development, implementation and application of the concept of 'diversity' to allow the search algorithm to generate designs which reflect different trade-offs without formulating a large number of rule-based objectives and constraints explicitly defining these trade-offs.

4. The development, implementation and application of a selection approach that is able to:

    - Support the transparent investigation of a large and diverse set of ship configurations to enable the naval architect to learn why particular designs are to be preferred over others, which is an insight crucial to settle trade-offs during the selection of designs.
    - Address the 'lack of completeness' by triggering the expression of human engineering judgement through the use of a filterable visualisation of the positioning of systems in the appropriate context: a general arrangement plan.
    - Generate acceptance by having the naval architect take all selection decisions. This instills a sense of ownership that is crucial to seeing a selected design through the remainder of the design process.

Novelty is important from a research point of view. However, it is not the only feature of the approach worth considering. Therefore, Section 8.6 will conclude this thesis by presenting some thoughts on the practical application of the approach. Before doing so, Section 8.5 will first discuss topics that warrant further investigation.

8.5 Future developments

The implementation of the packing and selection approaches were developed for the applications in Chapter 6 and are the result of an academic research effort. This means there is further scope for improvement to the approach and its implementation.

The following improvements to the approach are proposed:

- Developing the parametric model can only start after the ship's systems and the design requirements are available. Designing the systems and deriving requirements are both important (due to their impact on the resulting ship design, see Section 1.6), and time-consuming. Hence, support for this part of the design process is essential; it should built upon existing approaches, such as a 'functional decomposition' as proposed by *Wolff [140]*.

- The yield of the packing process can be improved by allowing objects to be split when the packing process fails for the original objects. The resulting set of more numerous but smaller objects should find more suitable positions to choose from and thus make it easier to comply with the overlap rules.

  However, the splitting of objects directly affects the clustering of systems (which results from human invention, see Section 4.4.1). Splitting, therefore, warrants careful consideration before it is integrated in the packing process as an additional design change for overlap management (Section 4.4.5).

- Several non-negotiable requirements concern buoyancy, stability and trim whose prediction relies heavily on the accurate estimation of both weight and centre of gravity. Section 5.3 explained, however, that no particular effort was made to develop an accurate weight and centroid estimate that is sensitive to both large and small changes in the ship configuration. Practical application of the approach developed in this dissertation does require the development of just such an accurate and sensitive weight estimate.

  One problem hampering the development of such accurate and sensitive weight and centre of gravity estimates is that the presence of some of the systems clustered in an object (Section 4.4.1) can depend on the location of that object in the ship. Assume, for example, that a single soft object represents an accommodation area and passageway. Such an object might weigh less if an area low in the ship is used as the basis for the weight estimate, or might weigh more when a similar area on main deck is used. The reason is that an accommodation area on main deck might have much more additional equipment (for damage control, for instance) and connections (ventilation ducts, cables, piping) enclosed in the same area, which will increase the weight of the object. Ideally, one should be able to distinguish between systems that do not change weight when changing positions, and those that do, when clustering systems in objects. This, however, requires the ships that are used as the basis for the weight estimate to be defined at a high level of detail.

- The non-negotiable requirements on stability and trim are currently safeguarded through constraints used by NSGA-II after completing the packing process. They could also be reformulated as constraints on positions which are considered during the packing process. This would, for example, prevent a heavy system of objects from taking a position in the ship that will shift the vertical centre of gravity upwards by such an extent that all systems packed afterwards cannot compensate for it (this situation would, therefore, result in an unstable and thus infeasible design).

This integration of centre-of-gravity-related constraints in the packing process is expected to improve yield considerably. It will reduce the number of constraints considered by NSGA-II and thereby result in a simpler search problem.

- Section 3.5.1 discussed the need to have a reusable parametric ship description. Though the two applications in Chapter 6 already illustrate that the object types, overlap rules and design changes of the packing approach were reusable for those particular applications, further investigation is warranted.

  In particular, the packing approach should be applied to other types of service vessels (e.g., a pipe-laying vessel, a submarine or a helicopter carrier) and other types of configuration-driven ships (such as chemical tankers) to study the extent with which it can cope with such vessels.

  Two recent applications took reusability one step further and investigated the ability of modified packing-based approaches to configure cargo in operational areas aboard service vessels (an amphibious assault ship in *Veldhuis et al. [134]* and a wind turbine installation vessel in *den Hamer [33]*, respectively). Such 'broader' applications also warrant further attention.

- The approach has only been applied to mono-hull type vessels. Further application to ship and offshore structures with more unconventional envelope shapes, e.g., jack-ups, semi-submersibles, catamarans, trimarans and SWATHs, is considered both possible and worthwhile. Among other things, it would provide more insight in the interaction between envelope shape and size and the interior arrangement.

  Still, application to unconventional envelope shapes will require considerable effort as one has to update weight and centre of gravity predictions (difficult enough for a mono-hull), as well as adjust numerous other performance prediction tools, e.g. stability, resistance, propulsion.

- The number of prediction tools integrated with the packing approach should increase. Even though the results do not have to be used by objectives or constraints, this will increase the information available to the naval architect to identify and resolve trade-offs during the selection process. Moreover, integrating more advanced prediction tools, e.g., *Raven [104]*, will help predict performances for a broader range of designs and, as a result, will improve the ability of the naval architect to further increase the competitiveness of service vessels.

  These additional prediction tools should, for warships at least, include sea-keeping (already used by *Van Bruinessen [119]*, but not integrated with the three-dimensional implementation of packing approach from Chapter 4) and vulnerability assessment (already developed in *Van Ingen [125]*, with a stand-alone interface to use designs generated by the packing approach).

- The approach outlined in Fig. 8.1 includes the possibility to use insight gained from the selection approach to adjust and improve the parametric ship description. Such feedback is greatly enhanced when the selection polygons (e.g., Fig. 7.9) can be used to define logical objects (Section 4.6). This will focus the parametric ship description towards more desirable configurations in subsequent search runs.

This feedback mechanism can be taken one step further by developing a truly interactive approach that mixes application of the packing-based ship description and the search algorithm with regular and intensive human interaction (similar to *Carlsen et al. [27]*; it also touches upon the 'sketch-based' approach advocated by *Pawling [95]* and *Pawling and Andrews [96]*).

In this interactive approach, a search algorithm would first generate numerous configurations using only a few systems and a few constraints, e.g., for their positions. From this set, the naval architect could select a small set of configurations deemed most suitable, 'freeze' their desirable features (while still allowing for some flexibility) and increase both the number of systems and the number of constraints. Then, the search process continues and could generate additional configurations (with the extra systems), while safeguarding the desirable features decided upon in the previous step. From this expanded set, the naval architect would again choose suitable designs that now would have both an increased number of systems and an increased number of desirable features that comply with an increasing number of constraints.

This process of steadily increasing both the number of systems and the number of constraints is repeated until all systems are included in the ship description, and all constraints are met, i.e., until complete feasible ship configurations are generated.

Such an interactive approach would enable the naval architect to gradually develop acceptable configurations that are both desirable and feasible, while still considering many alternative positions for each and every system in the configuration.

The following improvements to the implementation are proposed:

- The current implementation of the packing approach uses a slow but user-friendly programming language in Matlab for reasons outlined in Section 4.4.4. A lower-level programming language, e.g., FORTRAN, C++ or C# should increase the speed of computation and enable the search algorithm to investigate more alternatives in a given timeframe. Still, converting the implementation to a faster programming language should only be conducted when the implementation is considered to be mature enough.
- The current definition of systems of objects relies entirely on text-files. A graphical user-interface to construct a parametric ship description from the object types in Section 4.4.1 would improve the ease of use and increase the speed with which a suitable parametric ship description could be developed.

8.6 Thoughts on practical application of the approach

At the time of writing, the approach developed in this dissertation has been applied to four early stage design projects: the drillship (*Wagner [135]*) and frigate discussed in Chapter 6, a US Coastguard cutter application performed by *Van Bruinessen [119]* and an in-house application at the Defence Materiel Organisation in The Hague to study a mother-ship for stand-off mine clearance operations using manned and unmanned vehicles.

There are several observations from these applications:

- Both developing an accurate database of ship parts with shape, size and weight estimates at an appropriate level of detail and integrating these parts in a suitable parametric ship description is crucial.

  It also takes up a considerable amount of time, which makes careful planning of the early stage design process essential. Only this will ensure that the results are relevant, accurate and available in time to support the early stage design process.

  However, this problem related to preparation is not unique to the design approach presented in this dissertation. For example, such systems' shape, size and weight information must be produced for any early stage ship design project (as argued in Section 1.6 and in *Van der Nat [120]*).

- For some reason, naval architects tend to over-constrain the initial versions of the parametric ship description, up to the point that little diversity remains (later versions are more flexible, though).

  One possible cause of this could be their training. Currently, naval architects are trained to exert full control over the generation of designs; something purposely avoided in the approach developed in this thesis. This difference could lead to uncertainty over the outcome of the design process and, hence, a tendency to over-constrain the parametric ship description in an attempt to control it.

  Moreover, in the proposed approach, one must develop a parametric ship description that covers a wide range of configurations, i.e., in effect numerous ships have to be designed at the same time. This is a task quite different from developing the single concept that naval architects currently are trained to create.

- Application of the approach requires the naval architect to identify the relevant variations early on in the discussion with the customer, i.e., before developing the parametric ship description. Most importantly, he or she must establish what options are of interest (Section 1.3) and which requirements are negotiable and non-negotiable, respectively (Section 1.4). Failure to do this means the gains from using the packing-based design approach could well be negated by the irrelevancy of the results.

Even though the results from Chapter 6 are positive, the ultimate test of any computer-based design support is its practical application in a real-life ship design environment. At least one further application is planned at the Defence Materiel Organisation (in addition to the application to study a mother-ship for stand-off mine clearance operations mentioned above). This and other usages should give more insight in the practical utility of the approach.

Ultimately, time will tell...

Appendix A

Glossary

This dissertation uses a considerable number of terms in a somewhat peculiar way. Hence, a list of definitions is provided as reference for the reader. It explains each term in some detail and it also lists the page number where a particular term first is used in the main body of the text.

- **Acceptance**. At least some of the designs generated by a ship design process should prove to be acceptable to a naval architect. Such acceptance means that the naval architect has a sense of ownership for the ship design. The generation of acceptance is the only way that a naval architect with responsibility is willing to risk his / her reputation by using the design in the remainder of the design process. See page 42.
- **Aspects**, **characteristics**, **features**. Any property of a ship, e.g., a shape or position of system, the weight or centre of gravity of the ship, or a performance such as initial stability. Aspects, characteristics and features are used interchangeably. See pages 2, 5 and 16.
- **Believability**. When designs are considered believable by a naval architect (a term introduced by *Andrews [4]*), they are considered to 'make sense', to lack basic flaws and to be feasible (comply with a basic set of relevant non-negotiable requirements). See page 42.
- **Configuration**. An arrangement of systems with known relative positions that together describe the entire ship, i.e., hull, superstructure, as well as all equipment inside and on the ship. Configuration replaces the more common term arrangement, as the latter is deemed to concern the placement of equipment inside a fixed hull, while the approach in this dissertation seeks to change hull, superstructure and interior arrangement concurrently. See page 2.
- **Design requirement**. Required ability of the ship, drawn up by the naval architect and / or owner of the ship. Requirements can concern the ship's systems, its configuration and its performance levels. Moreover, some of the requirements will be non-negotiable and must therefore be met, whilst others are negotiable and may be traded off to meet other requirements. Also, not all design requirements will be available at the start of the design process, nor will all requirements be stated explicitly. See page 7.

- **Effectiveness**. A measure indicating the extent with which a ship is able to perform (part of) her mission. Which measure of effectiveness is used obviously depends on the mission the ship will perform. It may range from the percentage of aircraft shot down for a warship to the number of days a drillship can operate in a given area and season. The measure of effectiveness relates the effectiveness of a ship with that of her competitors, e.g., to other commercial ships operating in the same market, or to the threats imposed on a warship by the enemy. See page 3.

- **Engineering judgement**. Human knowledge gained by hand-on experience in design, engineering and production efforts. This kind of knowledge enables a naval architect to quickly assess whether a particular aspect of a ship's design makes sense, or whether it should be changed, without relying on elaborate performance predictions. See page 50.

- **Feasible ship design**. A ship design that meets a basic set of non-negotiable design requirements. In this thesis, this concerns the ability to float upright, for example. See page 8.

- **Focussing**. User-defined constraining that the naval architect can use to control changes in the parametric ship description. Helps to increase the yield of the search process. See page 114.

- **Function**. An activity that forms part of the ship's mission and that must be performed in order to execute and complete it. One example used in this dissertation is 'provide mobility', a function that -when fulfilled- enables the service vessel to transfer from port towards the relevant operating area at sea. See page 3.

- **Geometric packing problem**. The classic mathematical problem to place several geometric objects such that, first, all objects are enclosed in a larger positioning space and, second, no overlap exist among the objects. The geometrical packing problem is used as the basis for the parametric ship description developed in this dissertation. See page 61.

- **Implicit nature of trade-off decisions**. Trade-off decisions settle a conflict between design requirements by adjusting one or more of the requirements so that all requirements can be realised in a single ship design. Naval architects base their trade-off decisions on their opinions.

  Crucially, these opinions may change depending on the outcome of the trade-off decision, i.e., the ship design itself. Changing the opinion on which a trade-off decision is based obviously may cause changes to the trade-off decision itself. Hence, the trade-off decision remains subject to revision until all important consequences of the decision have become known, have been considered and were deemed to be acceptable by the naval architect.

  As such, initially, a trade-off decision is considered to be implicit, i.e., whether a trade-off decision needs adjustment, depends on the consequences of the trade-off decision, which in turn depend on the trade-off decision in the first place. The trade-off decision can be finalised once decision and consequences have been considered and brought in agreement with each other.

  Note, 'implicit' is used in the mathematic sense of the word, not in the more common meaning of something being 'unexpressed'. See page 22.

- **Input parameter**. A variable used by the search algorithm to alter a parametric description of the ship's design. It is invented by humans, who determine its type, bounds and meaning. Search algorithms only change its value. See page 31.

- **Invention**. A dedicated act of a human to create something that was not there before. Inventions range from the brilliant (inventing the steam engine) to the very mundane (writing a line of computer code). Regardless of their importance, inventions are considered to be a human premise, e.g., they cannot be made by a search algorithm or a computer-based parametric ship description. Instead, such software can merely use inventions made beforehand by humans, which caused the author to consider them to be unable to create something completely 'novel'. See pages 20 and 37.

- **Lack of completeness of ratings**. Search algorithms use rule-based ratings (objectives and constraints) to guide the search process. The ability of these ratings to identify desirable and undesirable designs is limited, e.g., due to their finite number. This means that a search algorithm must use an incomplete set of ratings to guide its search process and it is, therefore, not as comprehensive when rating a design as an experienced naval architect would be. Hence, search algorithms are deemed to suffer from a 'lack of completeness'. Note, increasing the number of ratings reduces the lack of completeness, but does not remove it. See pages 38 and 40.

- **Mission**. The operation the ship is to carry out at sea, e.g., dredge for sand and use it for land reclamation. Obviously, the mission differs per type of service vessel. Some service vessels may even need to be able to perform multiple missions either concurrently or sequentially. See page 2.

- **Object**. A cluster of systems that forms a single geometric entity (following *Van der Nat [120]*) that will be packed inside the positioning space. Clustering is performed by the naval architect prior to the packing process. Seven different object types are used: envelope, subdivision, hard, soft, free space, connection and logical. Together, the object types can describe the entire ship configuration. See pages 61 and 70.

- **Overlap management**. Overlap management is used to remove unwanted overlap between objects. To this end, both shape changes and position changes are used that translate the initial shape and position of an object that does not comply with the overlap rules, into a final shape and final position of an object such that it does comply with the overlap rules. See page 83.

- **Overlap rules**. Rules that define whether two objects may occupy the same part of the positioning space. Overlap rules are defined between pairs of object types. See pages 62 and 78.

- **Parametric ship description**. A description of the ship that can undergo changes by altering the value of input parameters. The parametric ship description presented in this dissertation uses a geometrical packing problem as its basis. It describes the ship (including hull and superstructure) as a collection of geometrical objects that are packed in a box-shaped positioning space. Input parameters changes the shape and / or size of objects as well as their initial positions. Overlap management ensures the objects are adjusted such that compliance with the overlap rules is achieved. Main benefit following from this packing-based approach is that the parametric ship description can undergo large and concurrent changes to the entire description of the ship driven by changes in input parameter values, which is something that existing approaches cannot do. See page 59.

- **Performance**. A technical ability of the ship, such as speed and initial stability. Typically measured in physical units relative to a neutral reference frame. Many performances can be predicted by performance prediction tools, i.e., computer-based mathematical models (such as *Holtrop and Mennen [60]* or *Raven [104]*). See page 3.

- **Positioning space**. The large box-shaped space (in this dissertation) in which objects are packed. The positioning space is larger than the largest object packed inside it (including the envelope). See pages 61 and 77.

- **Rating**. The search algorithm uses numerical objectives and constraints to guide its search effort. These objectives and constraints are called 'ratings' in this dissertation (derived from the verb 'to rate'), for two reasons. Firstly, the purpose of objectives and constraints is to rate designs according to their desirability, which allows the search algorithm to search for designs deemed more desirable. Secondly, in the approach developed in this thesis, the naval architect uses his / her engineering judgement after the search process to rate designs on aspects that were not considered during the search process by the numerical objectives and constraints. Hence, the word 'rating' is used both as a reminder of the real purpose of objectives and constraints, and to illustrate their relationship with other means to rate designs on their desirability, e.g., the use of engineering judgement. See pages 17 and 31.

- **Search algorithm**. A computer-based algorithm, commonly called 'optimisation algorithm', that changes the values of input parameters to search for improved values for the objective and constraint functions (objectives and constraints are called 'ratings' in this dissertation, see the term rating in this glossary). The term 'search algorithm' is used instead of 'optimisation algorithm' to avoid the common and overtly narrow impressions that in the practical sense of the word 'optimising' equals 'improving' and that 'optimal' equals 'best'. The correctness of both statements is highly dependent on the way an optimisation algorithm is used and the author therefore considers it to be safer to use search algorithm instead. Note, the term 'search algorithm' is also used for internet-based 'search engines', such as *Google*, but these obviously use a different kind of algorithm. See pages 17 and 29.

- **Service vessels**. A large class of ship types which all share both need and ability to use their 'cargo' to perform a mission at sea, e.g., drill for hydrocarbons. Service vessels differ in this respect from transport vessels, as the latter -predominantly- move cargo between ports. See page 1.

- **System**. A part of the ship. In this dissertation, any part of the ship is considered to be a system, regardless of its purpose or size. As such, the common distinction between systems, subsystems and components is not used in this dissertation; all parts are called systems instead. See page 2.

- **Systems of objects**. Sometimes, multiple objects are necessary to properly represent a cluster of systems (e.g., a gun might require both a 'hard object' representing the gun, and a 'free space object' representing the firing arc). These sets of multiple objects are packed concurrently and are therefore called 'systems of objects'. See page 101.

- '**Too optimal**'. The lack of completeness of ratings means the search algorithm cannot identify designs that are undesirable with regard to aspects for which no rule-based ratings are formulated. This unawareness is often exploited by search algorithms to improve ratings further at the cost of introducing basic flaws. Such designs are considered being 'too optimal' (following *Buonanno and Mavris [22]*), i.e., they are 'optimal' only for the ratings considered during the search process, and flawed in the practical sense with regard to one or more aspects not considered by the ratings used in the search process. One simple example is varying hull shape to reduce wave-resistance to the point where the hull becomes so slender that the ship will have a very low wave-resistance, but also will have become unstable. In summary, designs that are 'too optimal' are considered by the naval architect to be overtly optimal. See page 41.

- **Trade-off decision**. Some of the design requirements that the ship is to meet may conflict. This means it is not possible to create a ship design that satisfies both requirements simultaneously, e.g., because of an upper-bound on the available budget. In case of such a conflict, a trade-off decision is required by the naval architect that adjusts a requirement by allowing it to take priority over other requirements, i.e., one requirement is traded off against the other requirement to ensure the conflict is removed and designing a ship that meets all requirements becomes possible again. Also known as compromise decision. See page 21.

- **Transparency**. The ability to gain insight in interactions between different aspects and parts of the ship's design that result from a ship's integrated nature. Particularly necessary when studying a set of designs to identify conflicts between negotiable requirements. See page 41.

- **Yield**. The number of relevant alternatives found during the search process. It can be expressed, for example, as the percentage of designs that is feasible relative to the total number of designs considered during the search process. See page 37.

Appendix B

Exploded view of frigate design no. 36192 (test case no. 1)

The following Figures offer a three-dimensional view of the systems of objects that, together, define frigate design no. 36192 (test case no. 1) from Chapter 6.

- Fig. B.1: All systems
- Fig. B.2: Fight-related systems
- Fig. B.3: Move-related systems
- Fig. B.4: Hotel-related systems
- Fig. B.5: Accommodation-related systems
- Fig. B.6: Passageways and staircases routes

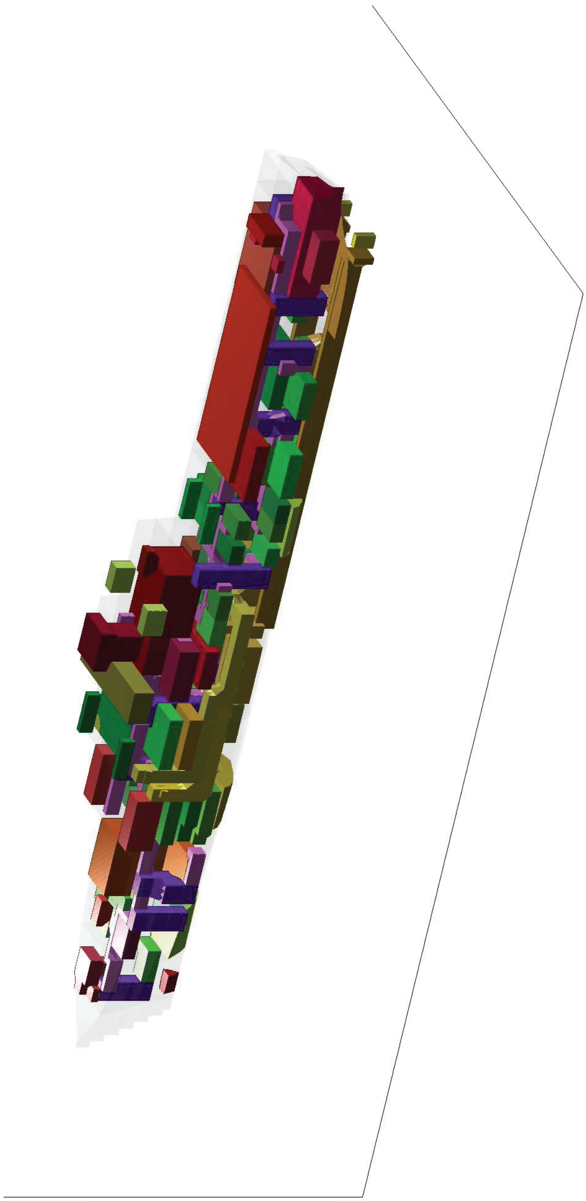


Fig. B.1 – Design no. 36192 (test case no. 1): all systems

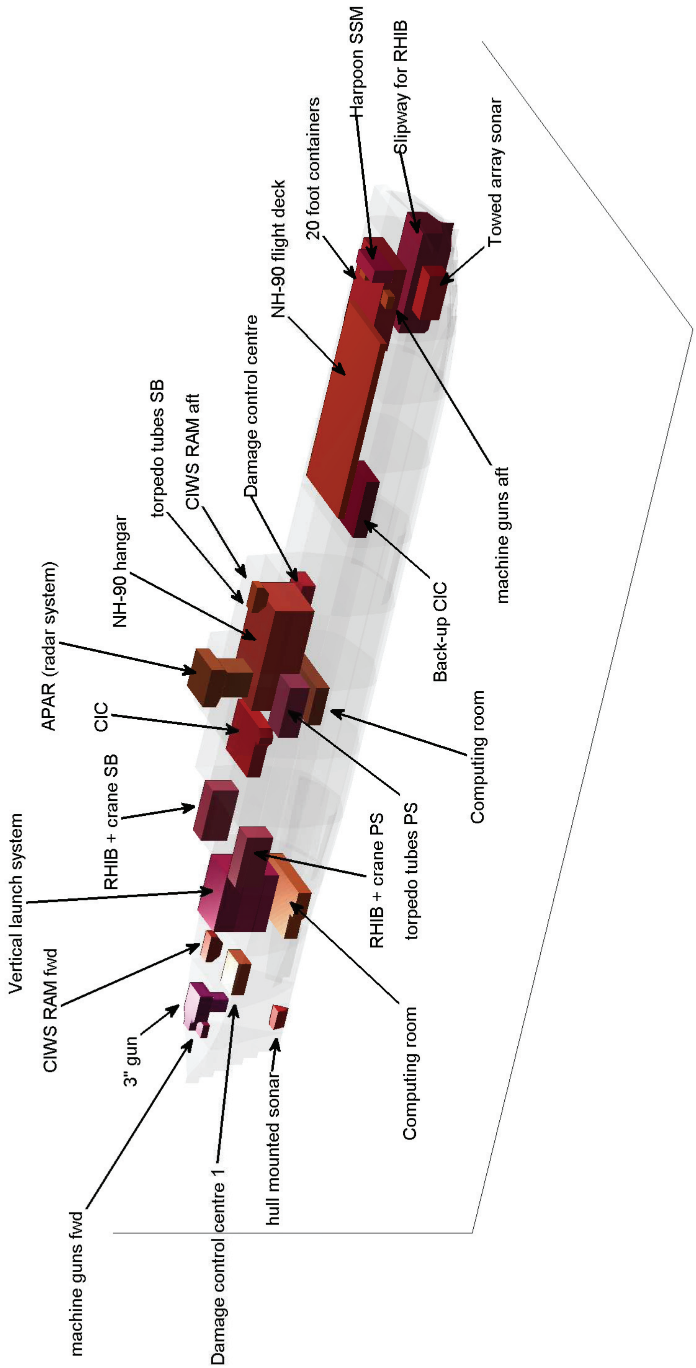


Fig. B.2 – Design no. 36192 (test case no. 1): fight-related systems

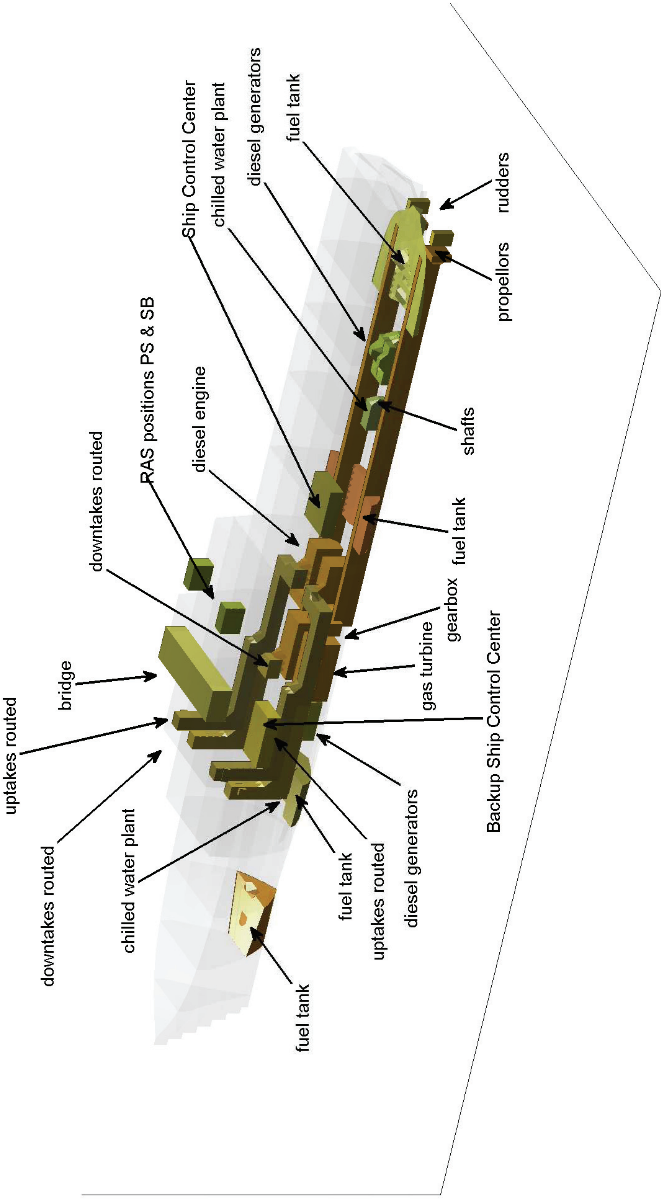


Fig. B.3 – Design no. 36192 (test case no. 1): move-related systems

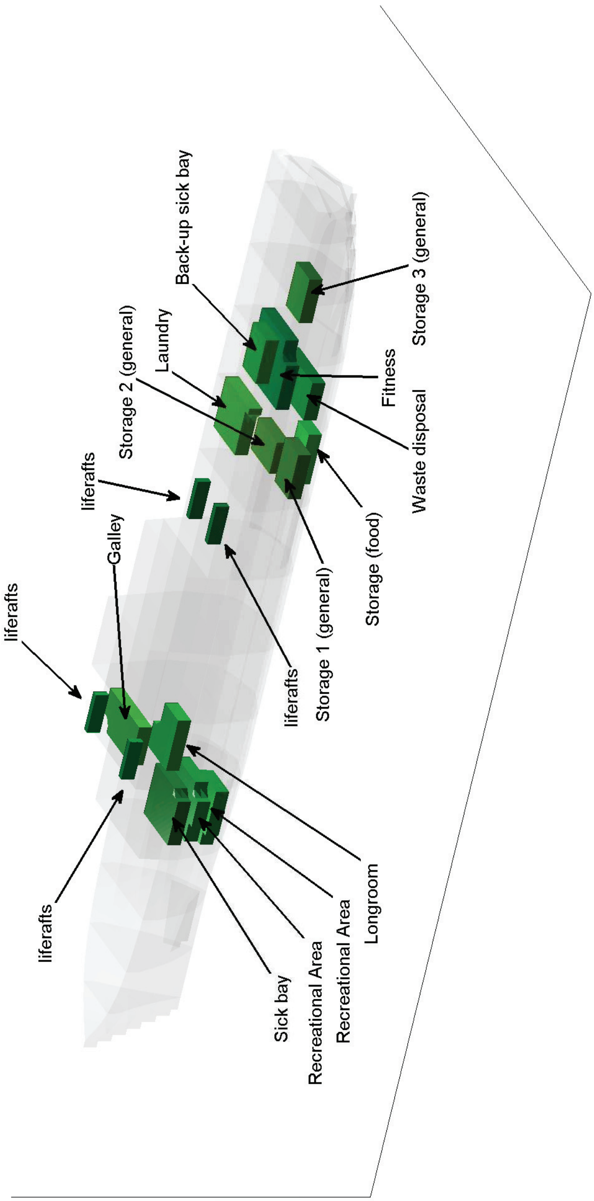


Fig. B.4 – Design no. 36192 (test case no. 1): hotel-related systems

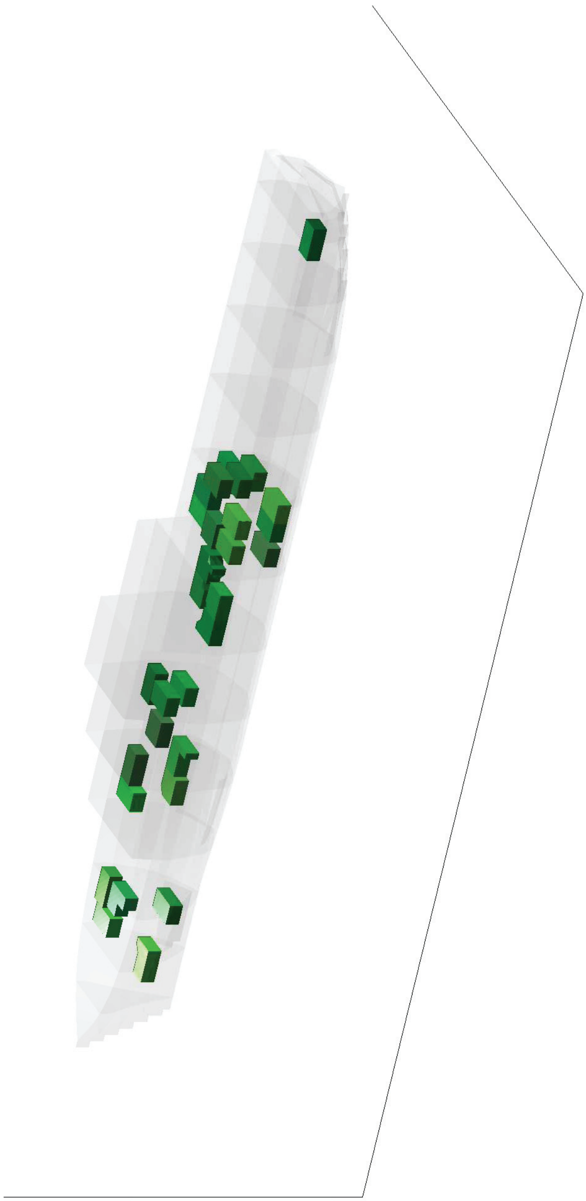


Fig. B.5 – Design no. 36192 (test case no. 1): accommodation (therefore, no labels)

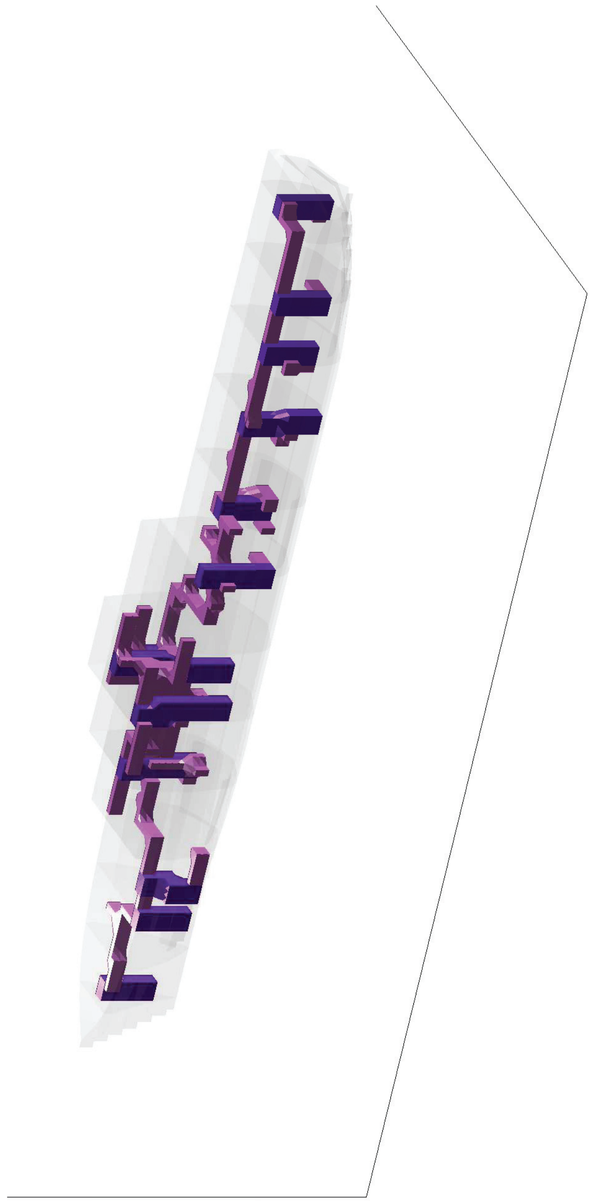


Fig. B.6 – Design no. 36192 (test case no. 1): passageways (pink) and staircases (purple)

Appendix C

Source code used for the applications in Chapter 6

The source code of the search algorithm, the packing-based parameteric ship description and the tools to process the resulting designs, are stored at Delft University of Technology at the Ship Design, Production and Operations group chaired by prof. ir. Hans Hopman.

Currently, these files are not freely available due to their proprietary nature. Still, one can make a request to Delft University of Technology to gain access to the source code for research and educational purposes. This is subject to approval from both Delft University of Technology and the author of this thesis.

Furthermore, commercial applications are encouraged, please contact either prof. ir. Hans Hopman at Delft University or the author for details.

Bibliography

[1] C. Abt, S. Harries, and K. Hochkirch. Constraint management for marine design applications. In *Proceedings of the Practical Design of Ship and Offshore Structures Conference (PRADS)*, Lübeck-Travemünde, Germany, September 2004.

[2] D. Andrews. Creative ship design. *Transactions of the Royal Institution of Naval Architects*, 123: 447–458, 1981.

[3] D. Andrews. An Integrated Approach to Ship Synthesis. *Transactions of the Royal Institution of Naval Architects*, 128:73–104, 1986.

[4] D. Andrews. A creative approach to ship architecture. *Transactions of the Royal Institution of Naval Architects*, 145(3), 2003.

[5] D. Andrews. Marine Design–Requirement Elucidation rather than Requirement Engineering. In *Proceedings of the 8th International Marine Design Conference*, pages 3–19, Athens, Greece, 2003.

[6] D. Andrews. Architectural considerations in carrier design. *International Journal of Maritime Engineering*, 146(4):227–246, 2004.

[7] D. Andrews and C. Dicks. The building block design methodology applied to advanced naval ship design. In *Proceedings of the International Marine Design Conference*, pages 3–19, Newcastle, United Kingdom, 1997.

[8] D. Andrews and R. Pawling. SURFCON, A 21^{st} century ship design tool. In *Proceedings of the 8th International Marine Design Conference*, Athens, Greece, 2003.

[9] D. Andrews and R. Pawling. Fast motherships - a design challenge. In *Proceedings of Warship 2004: Littoral Warfare & the Expeditionary Force*, London, United Kingdom, June 2004. Royal Institution of Naval Architects.

[10] D. Andrews, A. Cudmore, P. Humble, and D. Wilson. Subcon - a new approach to submarine concept design. In *Proceedings of Warship Symposium: Submarines 5*, London, United Kingdom, 1996. Royal Institution of Naval Architects.

[11] D. Andrews, A. Greig, and R. Pawling. The implications of an all electric ship approach on the configuration of a warship. In 7^{th} *International Naval Engineering Conference and Exhibition*, Amsterdam, The Netherlands, March 2004. The Institute of Marine Engineering, Science and Technology.

[12] A. Asmara and U. Nienhuis. Automatic piping system in ship. In *Proceedings of COMPIT 2006*, Oudepoelgeest, The Netherlands, April 2006.

[13] A. Asmara and U. Nienhuis. Automatic piping system implementation: A real case. In *Proceedings of COMPIT 2007*, Cortona, Italy, April 2007.

[14] M. Atiquzzaman, S. Liong, and X. Yu. Alternative decision making in water distribution network with NSGA-II. *Journal of water resources planning and management*, 132:122, 2006.

[15] P. Bernardoni, P. Bidaud, C. Bidard, and F. Gosselin. A new compliant mechanism design methodology based on flexible building blocks. *Smart Material and Structures*, 5383:244–254, 2004.

[16] C. Beyer, C. Chi, and M. Devine. User manual for ASSET/MONOSC: Advanced Surface Ship Evaluation Tool / Monohull Surface Combatant Program. Technical Report BCS 40530-10, David Taylor Research Center, 1990.

[17] V. Blouin, Y. Miao, X. Zhou, and G. Fadel. An assessment of configuration design methodologies. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, 2004. American Institute of Aeronautics and Astronautics.

[18] E. Boulougouris and A. Papanikolaou. Optimisation of Naval Ships with Genetic Algorithms for Enhanced Survivability. In *Proceedings of the 8th International Marine Design Conference*, pages 727–740, Athens, Greece, 2003.

[19] R. Brouwer. A Framework for Systems Engineering in Ship Design from a NATO Specialist Team Perspective. In *Proceedings of the Engineering the Total Ship Symposium*, Falls Church, United States, September 2008. American Society of Naval Engineers.

[20] A. Brown and J. Salcedo. Multiple-Objective Optimization in Naval Ship Design. *Naval Engineers Journal*, 115(4):49–61, 2003.

[21] D. Brown. *The future British surface fleet: Options for medium-sized navies*. Conway Maritime Press, 1991. ISBN 0851775578.

[22] M. Buonanno and D. Mavris. A new method for aircraft concept selection using multicriteria interactive genetic algorithms. In *Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, United States, January 2005. American Institute of Aeronautics and Astronautics.

[23] D. Burger and D. Horner. Design space exploration in the context of naval auxiliary ships. Technical report, QinetiQ Group PLC, 2010.

[24] I. Buxton. Engineering Economics Applied to Ship Design. *Transactions of the Royal Institution of Naval Architects*, 114:409–428, 1972.

[25] J. Cagan, K. Shimada, and S. Yin. A survey of computational approaches to three-dimensional layout problems. *Computer-Aided Design*, 34(8):597–611, 2002.

[26] Y. Cao. Efficient K-Nearest Neighbor Search using JIT, March 2008. URL `http://www.mathworks.com/matlabcentral/fileexchange/19345`.

[27] D. Carlsen, M. Malone, J. Kollat, and T. Simpson. Evaluating the Performance of Visual Steering Commands for User-Guided Pareto Frontier Sampling During Trade Space Exploration. In *Proceedings of ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 499–509, Brooklyn, New York, USA, 2008. American Society of Mechancial Engineers.

[28] A. Daniels and M. Parsons. Development of a Hybrid Agent-Genetic Algorithm Approach to General Arrangements. *Ship Technology Research*, 55(2):78–86, 2008.

[29] A. Daniels, F. Fahmashi, and D. Singer. Intelligent Ship Arrangement (ISA) Methodology - Improvements and Capability Enhancements. In *Proceedings of the 10th International Marine Design Conference*, pages 272–291, Trondheim, Norway, 2009.

[30] M. De Oliveira. Offshore platforms sizing optimization through genetic algorithms. In *Proceedings of Deep Offshore Technology International Conference*, Monaco, November 2008.

[31] K. Deb and R. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions On Evolutionary Computation*, 6(2):182–197, April 2002.

[33] T. den Hamer. Optimizing for design and operation - application of a configuration optimization routine to the design and operation of an offshore wind turbine installation jack-up. Master's thesis, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, April 2011.

[34] T. DeNucci and J. Hopman. Optimization-based approach to rationale capturing in ship design. In *Proceedings of COMPIT 2010*, Gubbio, Italy, April 2010.

[35] T. DeNucci, J. Hopman, and B. Van Oers. Capturing trade-off rationale during design selection. In *Proceedings of COMPIT 2009*, Budapest, Hungary, May 2009.

[36] T. DeNucci, B. Van Oers, and J. Hopman. Prototype development of a ship rationale capture tool. In *Proceedings of the International Conference of Ship and Shipping Research (NAV2009)*, Messina, Italy, November 2009.

[37] T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In *Proceedings of SIGGRAPH '98, the 25th annual conference on Computer graphics and interactive techniques*, pages 85–94, Orlando, Florida, United States, 1998. ACM.

[38] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1): 269–271, 1959.

[39] H. Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159, 1990.

[40] P. Dykstra and M. Muuss. The BRL-CAD package, an overview. In *USENIX, Proceedings of the Fourth Computer Graphics Workshop*, Cambridge, United States, 1987.

[41] M. Eames and T. Drummond. Concept Exploration-an Approach to Small Warship Design. *Transactions of the Royal Institution of Naval Architects*, 119, 1977.

[42] J. Eddy and K. Lewis. Multidimensional Design Visualization in Multiobjective Optimization. In *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 2002–5621, Atlanta, Georgia, United States, 2002. American Institute of Aeronautics and Astronautics.

[43] T. Eefsen, F. Van Walree, D. Peri, P. Van Terwisga, H. Kristensen, R. Dattola, and M. Visser. Development of frigate designs with good seakeeping characteristics. In *Proceedings of the 9th Symposium on Practical Design of Ships and Other Floating Structures, Luebeck-Travemuende, Germany*, 2004.

[44] N. Friedman. *Modern Warship: Design and Development*. Mayflower Books, 1979. ISBN 0831760826.

[45] V. Gantovnik, S. Tiwari, G. Fadel, and Y. Miao. Multi-objective vehicle layout optimization. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, United States, September 2006.

[46] A. Giassi, F. Bennis, and J. Maisonneuve. Multidisciplinary design optimisation and robust design approaches applied to concurrent design. *Structural and Multidisciplinary Optimization*, 28(5): 356–371, 2004.

[47] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 1989. ISBN 0201157675.

[48] B. Goris. Rapid Hull Modeling in Rhinoceros. *Rhino Centre Website*, 2005. `http://www.rhinocentre.nl/maritime_toepassingen.htm`.

[49] J. Govaarts. A method to identify size drivers of warships. Master's thesis, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, January 2011.

[50] C. Graham. The impact of subsystems on Naval Ship Design. *Naval Engineers Journal*, 87(6): 15–25, 1975.

[51] B. Grant. Density as a cost driver in naval submarine design and procurement. Master's thesis, Naval Post-Graduate School, Monterey, California, United States, June 2008.

[52] A. Greig, S. Rusling, and R. Bucknall. A Future RN fleet with an IFEP nuclear aircraft carrier. In *Proceedings of the* 9^{th} *International Naval Engineering Conference and Exhibition*, Hamburg, Germany, April 2008. The Institute of Marine Engineering, Science and Technology.

[53] P. Grignon. *Configuration design optimization method*. PhD thesis, Clemson University, 1999.

[54] P. Grignon and G. Fadel. A GA-based configuration design optimization method. *Journal of Mechanical Design*, 126:6–15, 2004.

[55] G. Grigoropoulos, T. Perdikari, V. Asouti, and K. Giannakoglou. MDO of Hull Forms Using Low-Cost Evolutionary Algorithms. In *Proceedings of the NATO-RTO-MP AVT-173 Workshop on Virtual Prototyping of Affordable Military Vehicles Using Advanced MDO*, Sofia, Bulgaria, 2011. North Atlantic Treaty Organisation. Downloadable after 1-1-2012 from `http://www.rto.nato.int/ACTIVITY_META.asp?ACT=AVT-173`.

[56] S. Harries. *Parametric Design and Hydrodynamic Optimization of Ship Hull Forms*. Mensch-und-Buch-Verlag, 1998. ISBN 3-933346-24-X.

[57] J. Hillen. New Defence minister takes office, October 2010. URL `http://www.defensie.nl/actueel/nieuws/2010/10/14/46174332/Nieuwe_minister_treedt_aan`.

[58] F. Hillier and G. Lieberman. *Introduction to operations research*. McGraw-Hill Science, Engineering & Mathematics, 2004. ISBN 007-123828-X.

[59] M. Hoekstra. *Numerical Simulation of Ship Stern Flows with a Space-Marching Navier-Stokes Method*. PhD thesis, Delft University of Technology, 1999.

[60] J. Holtrop and G. Mennen. An Approximate Power Prediction Method. *International Shipbuilding Progress*, 29(335):166–170, 1982.

[61] W. Hughes. A salvo model of warships in missile combat used to evaluate their staying power. *Naval Research Logistics (NRL)*, 42(2):267–289, 1995.

[62] G. Jacquenot. *Méthode générique pour l' optimisation d' agencement géométrique et fonctionnel*. PhD thesis, l' École Centrale de Nantes, 2010.

[63] A. Keane, W. Price, and R. Schachter. Optimization techniques in ship concept design. *Transactions of the Royal Institution of Naval Architects*, 133(part A):123–143, 1991.

[64] E. Keizer. "Future Reduced Cost Combatant study (NATO restricted)". Technical report, Directorate of Materiel, Royal Netherlands Navy, 1998.

[65] J. Kennedy, R. Eberhart, et al. Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, volume 4, pages 1942–1948, Piscataway, New Jersey, United States, 1995. IEEE.

[66] J. Keuning. " Grinding the bow" or" How to improve the operability of fast monohulls". *International Shipbuilding Progress*, 53(4):281–310, 2006.

[67] H.-S. Kim and S.-B. Cho. Application of interactive genetic algorithm to fashion design. *Engineering Applications of Artificial Intelligence*, 13(6):635–644, 2000.

[68] S. Kirkpatric, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598): 671–680, 1983.

[69] H. Klein Woud and D. Stapersma. *Design of Propulsion and Electric Power Supply Systems*. The Institute of Marine Engineering, Science and Technology, 2002.

[70] P. Kulkarni, S. Singh, and V. Seshadri. Parametric studies of exhaust smoke–superstructure interaction on a naval ship using CFD. *Computers and Fluids*, 36(4):794–816, 2007.

[71] R. Lamerton, N. Moss, M. R.E., and W. Ubhi. Propulsion options for future frigates, power dense solutions for medium sized warships. In *Proceedings of the* 9^{th} *International Naval Engineering Conference and Exhibition*, Hamburg, Germany, April 2008. The Institute of Marine Engineering, Science and Technology.

[72] K. Lee, S. Han, and M. Roh. Optimal Compartment Layout Design for a Naval Ship Using an Improved Genetic Algorithm. *Marine Technology*, 39(3):159–169, 2002.

[73] R. Leopold and W. Reuter. Three Winning Designs FDL, LHA, DD-963: Method and Selected Features. *Transactions of the Society of Naval Architects and Marine Engineers*, 79:297–365, 1971.

[74] A. Mahan. *Naval Strategy, Compared and Contrasted with the Principles and Practices of Military Operations on Land*. Little, Brown, and Company, 1911. `http://www.archive.org/stream/navalstrategyco00mahagoog#page/n6/mode/2up`.

[75] J. Maisonneuve, S. Harries, J. Marzi, H. Raven, U. Viviani, and H. Piippo. Towards optimal design of ship hull shapes. In *Proceedings of the 8th International Marine Design Conference*, pages 31–42, Athens, Greece, 2003.

[76] D. Malyna, J. Duarte, M. Hendrix, and F. Van Horck. Multi-objective optimization of power converters using genetic algorithms. In *Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM 2006)*, pages 713–717, 2006.

[77] P. Mandel and R. Leopold. Optimization Methods Applied to Ship Design. *Transactions of the Society of Naval Architects and Marine Engineers*, 74, 1966.

[78] E. Mayer and S. Cooper. Installed Power Corrupts: Speed and Naval Ship Design. In *Proceedings of the 9th International Marine Design Conference*, pages 499–509, Ann Arbor, United States, 2006.

[79] T. McDonald. *A Library Based Approach for Exploring Style in Preliminary Ship Design*. PhD thesis, University College London, 2010.

[80] R. McNeel and Associates. *Rhinoceros 4.0, NURBS modeling for Windows*, 2007. www.rhino3d.com.

[81] Y. Miao. *Packing Optimization of Engineering Problems*. PhD thesis, Clemson University, 2005.

[82] Y. Miao, G. Fadel, and V. Gantovnik. Vehicle configuration design with a packing genetic algorithm. *International Journal of Heavy Vehicle Systems*, 15(2):433–448, 2008.

[83] M. Mitchell. *An introduction to genetic algorithms*. The MIT press, 1998. ISBN 978-0262631853.

[84] Naval Sea Systems Command. NAVSEA Design Practices and Criteria Manual for General Arrangements Design, Chapter 070. Technical Report T9070-AB-PRO-010 Rev. A ed., Washington, DC: Naval Sea Systems Command, 1992.

[85] J. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4): 308, 1965.

[86] W. Nethercote and R. Schmitke. A Concept Exploration Model for SWATH Ships. *Naval Architect*, pages 113–130, 1982.

[87] W. Neu, W. Mason, S. Ni, Z. Lin, A. Dasgupta, Y. Chen, and V. Blacksburg. A Multidisciplinary Design Optimization Scheme for Containerships. In *Proceedings of the* 8^{th} *AIAA Symposium on Multidisciplinary Analysis and Optimization*, pages 6–8, Long Beach, California, United States, 2000.

[88] E. Nick. *Fuzzy Optimal Allocation and Arrangement of Spaces in Naval Surface Ship Design*. PhD thesis, University of Michigan, 2008.

[89] E. Nick and M. Parsons. Fuzzy Optimal Allocation of Spaces to Zone-decks in General Arrangements. In *Proceedings of the Practical Design of Ship and Offshore Structures Conference (PRADS)*, Houston, United States, 2007.

[90] E. Nick, M. Parsons, and B. Nehrling. Fuzzy Optimal Allocation of Spaces to Zone-decks in General Arrangements. In *Proceedings of the 9th International Marine Design Conference*, pages 651–671, Ann Arbor, United States, 2006.

[91] J. Nixon. *A Systematic Process for Adaptive Concept Exploration*. PhD thesis, Georgia Institute of Technology, 2006.

[92] P. O'Brien. *Technology and naval combat in the twentieth century and beyond*. Routledge, 2001. ISBN 978-0415449366.

[93] V. Pareto. Manuel d'economie politique. *Bull. Amer. Math. Soc. 18 (1912), 462-474. DOI: 10.1090/S0002-9904-1912-02237-1 PII: S*, 2(9904):02237–1, 1912.

[94] M. Parsons, H. Chung, E. Nick, A. Daniels, S. Liu, and J. Patel. Intelligent Ship Arrangements: A New Approach to General Arrangement. *Naval Engineers Journal*, 120(3):51–65, 2008.

[95] R. Pawling. *The Application of the Design Building Block Approach to Innovative Ship Design*. PhD thesis, University College London, 2007.

[96] R. Pawling and D. Andrews. Design Sketching - The Next Advance in Computer Aided Preliminary Ship Design? In *Proceedings of COMPIT 2011*, Berlin, Germany, May 2011.

[97] D. Peri and E. Campana. Multidisciplinary Design Optimization of a Naval Surface Combatant. *Journal of Ship Research*, 47(1):1–12, 2003.

[98] D. Peri and R. Dattola. Systematic Length Variation and Device Positioning Study for a frigate ship accounting propulsion and operability characteristics. In *Proceedings of the International Conference on Ship and Shipping Research (NAV2003)*, Palermo, Italy, 2003.

[99] L. Piegl and W. Tiller. *The NURBS book*. Springer Verlag, 1997. ISBN 978-3540615453.

[100] C. Pouw. Development of a multiobjective design optimization procedure for marine propellers. Master's thesis, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, December 2007.

[101] J. Pugh and A. Martinoli. Discrete multi-valued particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium*, volume 6, pages 103–110, 2006.

[102] M. Purvis. Post-War RN Frigate and Guided Missile Destroyer Design 1944-1969. *Transactions of the Royal Institution of Naval Architects*, 116:189–206, 1974.

[103] K. Rasheed and H. Hirsh. Learning to be selective in genetic-algorithm-based design optimization. *AI EDAM*, 13(03):157–169, 1999.

[104] H. Raven. *A solution method for the nonlinear ship wave resistance problem*. PhD thesis, Delft University of Technology, 1996.

[105] P. Rigo. Least-Cost Structural Optimization Oriented Preliminary Design. *Journal of Ship Production*, 17(4):202–215, 2001.

[106] H. Rittel and M. Webber. Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169, 1973.

[107] T. Sarchin and L. Goldberg. Stability and buoyancy criteria for US naval surface ships. *Transactions of the Society of Naval Architects and Marine Engineers*, 70:418–458, 1962.

[108] P. Sen and J. Yang. Multiple-criteria Decision-making in Design Selection and Synthesis. *Journal of Engineering Design*, 6(3):207–230, 1995.

[109] T. Simpson, D. Spencer, M. Yukish, and G. Stump. Visual steering commands and test problems to support research in trade space exploration. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia, Canada, 2008.

[110] N. Smith, W. Hills, and G. Cleland. A layout design system for complex made-to-order products. *Journal of Engineering Design*, 7(4):363–375, 1996.

[111] J. Stepanchick and A. Brown. Revisiting DDGX/DDG-51 concept exploration. *Naval Engineers Journal*, 119(3):67–88, 2007.

[112] G. Stump, M. Yukish, T. Simpson, and L. Bennett. Multidimensional Visualization and Its Application to a Design by Shopping Paradigm. In *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 4–6, Atlanta, Georgia, United States, 2002. American Institute of Aeronautics and Astronautics.

[113] E. Takken. Concept design by using functional volume blocks with variable resolution. Master's thesis, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, May 2009.

[114] The Mathworks Inc. *Matlab*. Natick, Massachusetts, United States, 2010b edition, 2010.

[115] S. Tiwari, P. Koch, G. Fadel, and K. Deb. AMGA: an archive-based micro genetic algorithm for multi-objective optimization. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 729–736. ACM, 2008.

[116] J. Tuitman. *Hydro-elastic response of ship structures to slamming induced whipping*. PhD thesis, Delft University of Technology, 2010.

[117] F. Valdenazzi, S. Harries, C. Janson, M. Leer-Andersen, J. Maisonneuve, J. Marzi, and H. Raven. The FANTASTIC RoRo: CFD optimisation of the forebody and its experimental verification. In *Proceedings of the International Conference on Ship and Shipping Research (NAV2003)*, Palermo, Italy, 2003.

[118] S. Valliyappan and T. Simpson. Exploring Visualization Strategies to Support Product Family Design Optimization. In *Proceedings of the 11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 1–14, Portsmouth, Virginia, United States, 2006. American Institute of Aeronautics and Astronautics.

[119] T. Van Bruinessen. Concept exploration during the design exploration phase. Master's thesis, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, May 2010.

[120] C. Van der Nat. *A Knowledge-based Concept Exploration Model for Submarine Design*. PhD thesis, Delft University of Technology, 1999.

[121] C. Van der Nat, H. Klein Woud, and D. Stapersma. Conceptual submarine design. How to achieve an efficient and flexible design system. *Proceedings of IMarEST, Part B - Journal of Marine Design and Operations*, 2005(8):39–48, 2005.

[122] M. Van Diessen. The implementation of a bulkhead positioning routine in a conceptual design model. Technical report, Netherlands Defence Academy, October 2007.

[123] C. Van Es and M. Van Hees. Application of knowledge management in the conceptual naval ship design. In *Proceedings of COMPIT 2003*, Hamburg, Germany, 2003.

[124] M. Van Hees. *QUAESTOR: Expert governed parametric model assembling*. PhD thesis, Delft University of Technology, 1997.

[125] G. Van Ingen. An methodology to reduce warship vulnerability (in dutch: Een methodiek voor het reduceren van de kwetsbaarheid van marineschepen). Master's thesis, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, January 2011.

[126] W. Van Lammeren, D. Manen, M. Oosterveld, and N. Basin. *The Wageningen B-screw series*. Netherlands Ship Model Basin, 1969.

[127] B. Van Oers. Designing the process and tools to design affordable warships. In *Proceedings of the NATO-RTO-MP AVT-173 Workshop on Virtual Prototyping of Affordable Military Vehicles Using Advanced MDO*, Sofia, Bulgaria, 2011. North Atlantic Treaty Organisation. Downloadable after 1-1-2012 from `http://www.rto.nato.int/ACTIVITY_META.asp?ACT=AVT-173`.

[128] B. Van Oers and D. Stapersma. Applying first-principle tools in naval ship design. In *Proceedings of Warship 2006: Future Surface Warships*, London, United Kingdom, June 2006. Royal Institution of Naval Architects.

[129] B. Van Oers, D. Stapersma, and J. Hopman. Development and implementation of an optimisation-based space allocation routine for the generation of feasible concept designs. In *Proceedings of COMPIT 2007*, Cortona, Italy, April 2007.

[130] B. van Oers, D. Stapersma, and J. Hopman. Issues when selecting naval ship configurations from a pareto-optimal set. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia, Canada, September 2008.

[131] B. Van Oers, M. Van Hees, D. Stapersma, and H. Hopman. Combining a Knowledge System with Computer-Aided Design. *Ship Technology Research*, 55(2):51–59, 2008.

[132] B. Van Oers, D. Stapersma, and J. Hopman. An optimisation-based space allocation routine for the generation of feasible ship designs. *Ship Technology Research*, 56(1):31–48, 2009.

[133] B. Van Oers, D. Stapersma, and J. Hopman. A 3D packing approach for the early stage configuration design of ships. In *Proceedings of the* 10^{th} *International Naval Engineering Conference and Exhibition*, Portsmouth, United Kingdom, May 2010. The Institute of Marine Engineering, Science and Technology.

[134] D. Veldhuis, H. Koelman, and B. Nienhuis. Developing feasible stowage plans by means of an evolutionary algorithm. In *Proceedings of the 10th International Marine Design Conference*, pages 424–434, Trondheim, Norway, 2009.

[135] K. Wagner. Applicability of a configuration optimisation routine to the conceptual design of deepwater drilling vessels. Master's thesis, Delft University of Technology, Department of Marine and Transport Technology, Delft, The Netherlands, April 2009.

[136] K. Wagner, B. Van Oers, A. Wassink, and J. Hopman. Practical application of early stage ship configuration optimization: Deepwater drillship design. In *Proceedings of the Practical Design of Ship and Offshore Structures Conference (PRADS)*, Rio de Janeiro, Brazil, September 2010.

[137] K. Wagner, A. Wassink, B. Van Oers, and J. Hopman. Modeling complex vessels for use in a 3d packing approach: An application to deepwater drilling vessel design. In *Proceedings of COMPIT 2010*, Gubbio, Italy, April 2010.

[138] D. Watson. *Practical ship design*. Elsevier Science Ltd., 2002. ISBN 978-0080429991.

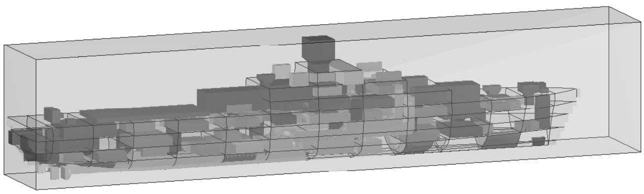
[139] P. Wolff. The development of a remote controlled mine sweeper. In *Proceedings of INEC 1994: Cost effective maritime defence*, Plymouth, United Kingdom, 1994.

[140] P. Wolff. *Conceptual design of warships*. PhD thesis, University of Twente, 2000.

[141] L. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30(3):407–428, 1975.

Summary

A packing approach for the early stage design of service vessels



Over the past decades, European shipyards and ship design firms have specialised successfully in the design and production of service vessels, a broad class of ships which perform complex operations at sea. Common examples of service vessels are dredgers, cruise ships, frigates, drillships, crane ships, amphibious assault ships and pipe-laying vessels.

Designing a competitive service vessel is not straightforward. The ship has to have relevant systems, which must be integrated into a coherent configuration and the ship as a whole must have appropriate performances. It is up to the naval architect to identify -very early on- which particular combinations of systems, configuration and performance might be of interest to the ship's owner. Doing so requires the naval architect to thoroughly investigate a large number of potentially interesting alternatives, which, unfortunately, takes a considerable effort. This is problematic due to both the large number of potentially interesting alternatives and the limited time that is available in early stage ship design. As a result, naval architects currently have to consider only a few alternatives and have to resort to rules-of-thumb and assumptions to make up for the limited numbers.

This inability to investigate a large number of alternatives early on is considered hampering the design of competitive service vessels. For this reason, this thesis presents a novel design approach that significantly increases the number of alternatives that naval architects can investigate during early stage design. The approach is based around the use of an existing search algorithm (commonly called 'optimisation algorithm') integrated with a novel 'packing-based' parametric ship description. Together, they can generate a large and diverse set of three-dimensional ship designs without human interaction. Furthermore, they can ensure that all designs comply with a basic set of non-negotiable design requirements.

This approach has two perceived benefits. Firstly, being able to generate designs automatically significantly increases the number of three-dimensional ship designs that can be created in a given time-frame. The naval architect can use these designs to study which combinations of systems, configuration and performances are promising. Secondly, generating feasible designs relieves the naval architect from having to safeguard important non-negotiable requirements, e.g., be able to float in upright condition. This enables the naval architect to focus on identifying and selecting those alternatives deemed to be most promising. The approach consists of three elements: the parametric ship description, the search algorithm and the selection approach.

The parametric ship description forms the core of the design approach as it both describes and alters the ship's systems and their configuration. It is based on a classic group of mathematical problems called 'packing problems'. It models the ship as a collection of geometrical objects that together represent the ship's systems (seven object types are used: 'envelope', 'subdivision', 'hard', 'soft', 'free space', 'connection' and 'logical'). Note, the packing-based approach assumes that the ship's systems are known beforehand, even though creating them is a design process in itself. These objects are packed in a boxed shape positioning space, which is larger than the ship's dimensions (as shown on the previous page). A simple set of overlap rules defines whether objects may overlap, which is crucial to generate a coherent configuration. Overlap management, i.e., changes in position and / or shape of objects, ensures that the overlap rules can be enforced. Packing the objects takes place in a user-defined sequence and continues until, either an object cannot be packed in compliance with the overlap rules, or until all objects are packed successfully. If all objects are packed successfully, performances can be predicted.

Key features of the packing-based ship description are, firstly, the ability to handle large and concurrent changes to the entire ship design that are driven by changes in input parameter values, secondly, the ability to model both weight-driven and space-driven ships at a variable level of detail and, thirdly, its reusability, which means it can be applied to different types of service vessels.

A genetic search algorithm, NSGA-II, uses the packing-based description to search for feasible ship designs. It uses a single objective: packing density, to search for dense designs, and several constraints that ensure designs comply with basic non-negotiable requirements (e.g., initial stability, speed and endurance). Most importantly, no attempt was made to formulate a 'complete' rule-based set of objectives and constraints since the effort required to develop a set that can evaluate a ship design in a manner as comprehensive as an experienced naval architect was considered being prohibitive. As important, no objectives were formulated to handle trade-offs between conflicting aspects of the design. The reason is that such trade-off decisions are -at the start of the design process- considered being 'implicit', i.e., they could need adjustment if their impact on the design is considered unacceptable by the naval architect. Hence, it is difficult to establish a-priori what a suitable compromise will look like. As a result, formulating rule-based objectives that take trade-off decisions was deemed not worthwhile. Using such objectives would also fail to generate acceptance, i.e., instill a sense of ownership for the resulting designs. The decisions had three consequences: the generation of designs that may contain flaws due to insufficient constraining, a limited ability to distinguish between designs, and the need to settle trade-offs without using rule-based objectives. These consequences are addressed in an a-posteriori manner by the selection approach.

Selecting designs is the last element of the approach. It helps the naval architect to study the large set of feasible designs, and to identify and select those designs deemed most promising in a transparent manner. Support is offered in several ways. Firstly, studying all feasible designs on all relevant aspects (regardless of their objective values) helps the naval architect to learn why some designs are to be preferred over others. Secondly, gaining such insight is, however, possible only through the provision of filtering, which prevents information overflow. Thirdly, expressing engineering judgement, triggered by visualising system positions in a general arrangement plan, increases the ability to distinguish between designs. It also helps to identify flaws caused by insufficient constraining, which thus increases believability.

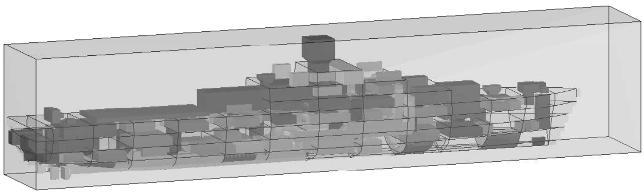
The naval architect uses this insight to establish the consequences of different trade-off decisions in a rapid decide-consider-revise cycle. Selection decisions are taken by the naval architect and are expressed through simple hand-drawn polygons. Subsequently, the polygons are used to identify and remove non-compliant designs, which helps to work out the consequences the decision has for the resulting designs. In turn, this insight in the consequences helps the naval architect to establish whether a decision needs to be revised. In particular, the ability to take, consider and revise decisions quickly, helps to remove the implicit nature of trade-off decisions. Most importantly, having a naval architect select designs based on both their objective values and his / her own engineering judgement generates the crucial acceptance for the designs that are selected.

The packing-based ship description, search algorithm and selection approach were applied in two test cases; one dealt with frigates, the other with drillships. The two applications showed that the resulting design approach is indeed able to generate a large and diverse set of feasible three-dimensional ship designs. Moreover, the approach proved capable of studying the impact of the type of changes in design requirements a naval architect might encounter during the design of service vessels. Examples included changes in endurance of a frigate, as well as changes to her weapon and sensor fit. Lastly, one of the sets of frigate designs was used to illustrate the workings of the selection approach by investigating how selecting designs that have particular features (e.g., better operability) affects the displacement and size of the resulting ships.

In summary, the novel approach presented in this thesis reduces the effort required to generate a large and diverse set of feasible, three-dimensional ship designs. It helps the naval architect to investigate them, and, ultimately, assists in the selection of those designs deemed most promising by the naval architect. As such, the novel approach is expected to improve the naval architect's ability to identify promising combinations of systems, configuration and performance early on, which should lead to the design of more competitive service vessels.

Samenvatting

Een op 'packing'-problemen gebaseerde methodiek voor het conceptontwerp van complexe werkschepen



De afgelopen decennia hebben Europese werven en ontwerpbureaus zich gespecialiseerd in het ontwerpen en de productie van complexe werkschepen ('service vessels'). Deze schepen kenmerken zich doordat ze op zee veeleisende operaties uitvoeren. Typische voorbeelden van deze schepen zijn baggerschepen, cruise schepen, fregatten, boorschepen, kraanschepen, amfibische transportschepen en pijpenleggers.

Een succesvol en winstgevend werkschip ontwerpen is echter verre van eenvoudig. Het schip moet alle relevante systemen aan boord hebben, waarbij deze systemen bovendien goed geïntegreerd moeten zijn in een doordachte indeling. Verder dient het schip als geheel goed te presteren door bijvoorbeeld de vereiste werkbaarheid te hebben in een bepaalde zeetoestand.

Het is dan ook de taak van de ontwerper om in een vroeg stadium veelbelovende combinaties van systemen, indeling en prestaties te identificeren. Dit vereist dat de ontwerper een groot aantal mogelijk interessante alternatieven grondig kan onderzoeken, hetgeen momenteel veel tijd kost. Dit brengt een probleem met zich mee omdat er tijdens de conceptontwerpfase onvoldoende tijd is om dit te doen. Het gevolg is dat ontwerpers tegenwoordig slechts enkele scheepsontwerpen creëren tijdens het conceptontwerp en het beperkte inzicht verkregen uit deze ontwerpen aan moeten vullen met vuistregels en ervaring. Dat het thans onpraktisch is om tijdens het conceptontwerp veel verschillende alternatieven te creëren, wordt door de auteur als problematisch beschouwd; het staat namelijk het ontwerpen van een succesvol en winstgevend werkschip in de weg.

Dit proefschrift presenteert daarom een nieuwe ontwerpmethodiek die de ontwerper in staat stelt tijdens het conceptontwerp in korte tijd veel meer alternatieven te genereren en te bestuderen. De aanpak is gebaseerd op de toepassing van een bestaand zoekalgoritme (ook wel bekend als optimalisatiealgoritme) in combinatie met een op packing-problemen gebaseerd parametrisch model van het schip. Samen kunnen ze, zonder interactie met de ontwerper, een grote en diverse verzameling aan driedimensionale scheepsontwerpen genereren. Bovendien is de methodiek in staat om te garanderen dat alle ontwerpen voldoen aan belangrijke haalbaarheidseisen, zoals het rechtop kunnen drijven in stabiele toestand.

Deze methodiek heeft twee voordelen. Allereerst geeft het automatisch genereren van driedimensionale scheepsontwerpen de mogelijkheid om het aantal alternatieven wat de ontwerper kan bestuderen sterk uit te breiden. Met deze grote verzameling alternatieven kan de ontwerper vervolgens inzicht krijgen in welke combinaties van systemen, indeling en prestaties veelbelovend en uiteindelijke winstgevend zijn. Ten tweede, doordat de haalbaarheid van de ontwerpen verzekerd is, heeft de ontwerper de vrijheid om zich te richten op het vinden van de meest interessante haalbare concepten.

De nieuwe ontwerpmethodiek bestaat uit een drietal onderdelen: een parametrische beschrijving van het schip, het zoekalgoritme en de methode om ontwerpen te selecteren.

De parametrische beschrijving van het schip vormt de kern van de ontwerpmethodiek en zorgt voor de beschrijving en variatie van zowel de systemen van het schip als van de indeling ervan. De parametrische beschrijving is gebaseerd op een bekende klasse van wiskundige problemen geheten packing-problemen.

De systemen van het schip worden gemodelleerd door een verzameling geometrische objecten. De parametrische beschrijving gebruikt in totaal zeven typen objecten, zijnde: “envelope”, “subdvision”, “hard”, “soft”, “free space”, “connection” en “logical”. Hierbij is aangenomen dat de systemen zowel ontworpen als bekend zijn aan het begin van het scheepsontwerpproces. Voorgenoemde geometrische objecten worden vervolgens geplaatst in een blokvormige positioneringsruimte, zoals getoond in de afbeelding op de vorige pagina. Deze positioneringsruimte is groter dan de maximum afmetingen van het schip, hetgeen essentieel is om variaties in scheepsgrootte mogelijk te maken.

Overlapregels beschrijven welke objecten dezelfde plaats in de positioneringsruimte mogen innemen (m.a.w. mogen overlappen), en welke niet. Om te zorgen dat alle objecten voldoen aan de overlapregels kan door middel van “overlap management” de positie en / of de vorm van objecten worden aangepast. De objecten worden sequentieel geplaatst in een door de ontwerper bepaalde volgorde.

Het plaatsen van objecten gaat door totdat alle objecten geplaatst zijn, of totdat er voor een object geen geschikte positie en / of vorm gevonden kan worden waarmee aan de overlapregels kan worden voldaan. In het laatste geval zal het plaatsingsproces stoppen. De beschrijving van het schip is compleet als alle objecten geplaatst zijn, waarna met voorspellingsmodellen prestaties zoals stabiliteit en behaalde snelheid berekend worden.

Het op packing-problemen gebaseerde parametrische model van het schip heeft een drietal belangrijke voordelen. Allereerst kan de beschrijving grote wijzigingen aan het gehele ontwerp van het schip teweegbrengen door de waardes van parameters te laten veranderen. Ten tweede kan de beschrijving overweg met verschillende detailniveaus, en kan deze zowel gewichts- als ruimtegedreven ontwerpen aan. Tot slot is de beschrijving herbruikbaar en kan worden toepast voor het ontwerpen van verschillende scheepstypen.

Het tweede onderdeel van de methodiek is het zoekalgoritme, een optimalisatiealgoritme genaamd NSGA-II. Dit algoritme gebruikt de parametrische beschrijving van het schip om naar haalbare ontwerpen te zoeken. Hierbij wordt slechts 1 doelfunctie gebruikt, zijnde de vullingsgraad. Daarnaast worden een aantal constraints gebruikt, welke ervoor zorgen dat de ontwerpen voldoen aan belangrijke haalbaarheidscriteria met betrekking tot stabiliteit, snelheid en endurance.

Er is met nadruk geen poging gedaan om een "complete" verzameling van doelfuncties en constraints te formuleren. Reden hiervoor is dat het zeer tijdrovend is om een dergelijke verzameling van doelfuncties en constraints te formuleren die net zo veelomvattend zal zijn als de beoordeling van een ervaren scheepsontwerper. Even belangrijk is dat er gekozen is om geen doelfuncties te formuleren die compromissen sluiten tussen conflicterende eisen aan het ontwerp. De reden hiervoor is dat dergelijke compromissen aan het begin van het ontwerpproces als "impliciet" worden beschouwd. Het wordt namelijk door de auteur niet mogelijk verondersteld om voordat een ontwerp is gemaakt volledig inzicht te krijgen in de gevolgen van de beslissing. Bovendien creëert het ontwerpproces nieuwe inzichten waardoor een compromis dat eerst wenselijk lijkt, later minder wenselijk blijkt. Beiden zaken hebben ertoe geleid dat het vooraf -a priori- formuleren van doelfuncties om compromissen te sluiten als niet de moeite waard wordt beschouwd. Het gebruik van voornoemde doelfuncties heeft bovendien nog een ander belangrijk nadeel. Ze zorgen er namelijk voor dat de beslissingen die het zoekalgoritme er op baseert niet geaccepteerd zullen worden door de ontwerper, omdat deze er geen invloed en controle over heeft gehad.

De keuze om 1 doelfunctie en slechts enkele constraints te gebruiken veroorzaakt drie belangrijke problemen. Allereerst kan het zoekproces leiden tot ontwerpen die fouten bevatten veroorzaakt door een gebrek aan constraints. Ten tweede heeft het zoekalgoritme maar beperkte mogelijkheden om verschillen tussen goede en slechte ontwerpen te identificeren. Ten derde moet er een alternatief worden gevonden om compromissen tussen conflicterende eisen aan het ontwerp te sluiten. Deze drie problemen worden opgelost door de selectieaanpak waarmee de ontwerper nadat het zoekproces is afgerond -a posteriori- scheepsontwerpen kan onderzoeken en kiezen.

De ondersteuning voor het kiezen van een geschikt ontwerp vormt het derde en laatste onderdeel van de ontwerpmethodiek. De ondersteuning helpt de scheepsontwerper op een aantal manieren om de grote verzameling haalbare ontwerpen te onderzoeken, waarna hij vervolgens met dit inzicht de meest veelbelovende ontwerpen kan identificeren en selecteren. De ontwerper kan ondermeer alle relevante aspecten van de gevonden haalbare scheepsontwerpen onderzoeken. Dit geeft hem het inzicht waarom sommige ontwerpen beter zijn dan andere. Hierbij is het filteren van de grote hoeveelheid gegevens van essentieel belang, aangezien dit ervoor zorgt dat de ontwerper het overzicht niet verliest. Daarnaast wordt de ontwerper gestimuleerd om, door middel van visualisatie van de indeling, het scheepsontwerp te beoordelen op basis van eigen ervaring, zodat daarmee de zeer beperkte beoordeling van het ontwerp door de doelfuncties en constraints uitgebreid kan worden. Dit geeft bovendien de mogelijkheid om fouten in het ontwerp te ontdekken en weg te nemen, hetgeen de geloofwaardigheid en daarmee de acceptatie van het gekozen scheepsontwerp vergroot.

Met het verkregen inzicht kan de ontwerper vervolgens de gevolgen van verschillende compromisbeslissingen onderzoeken. Dit gebeurt door snel compromisbeslissingen te nemen, de gevolgen ervan te bestuderen en, indien noodzakelijk, de compromisbeslissingen aan te passen. De ontwerper neemt zelf alle compromisbeslissingen en drukt deze uit door een polygoon te tekenen die wenselijke ontwerpen omsluit. Deze polygonen worden vervolgens gebruikt om de scheepsontwerpen die niet aan de keuze van de ontwerper voldoen te verwijderen, hetgeen helpt inzicht te krijgen in de gevolgen die de compromisbeslissing heeft voor het uiteindelijk ontwerp. Naar aanleiding hiervan beoordeelt de ontwerper in hoeverre de gevolgen acceptabel zijn, dan wel of dat de compromisbeslissing aangepast moet worden.

Deze snelle cyclus van beslissen, beoordelen en aanpassen geeft de scheepsontwerper de mogelijkheid om verschillende compromissen snel te bestuderen en neemt daarmee de impliciete eigenschap van compromisbeslissingen weg. Bovendien zorgen het beoordelen van ontwerpen op basis van ervaring, én de verantwoordelijkheid van de ontwerper om zelf de ontwerpen te kiezen, er samen voor dat de gekozen ontwerpen geloofwaardig en acceptabel zullen zijn.

De parametrische beschrijving van het schip, het zoekalgoritme en de selectieaanpak zijn toegepast om twee verschillende soorten werkschepen te ontwerpen. Als eerste testcase is voor een fregat gekozen. De tweede testcase betreft een boorschip waarmee op diep water naar olie en gas kan worden geboord. Beide testcases laten zien dat de nieuwe ontwerpmethodiek daadwerkelijk in staat is om een grote en diverse verzameling aan haalbare, driedimensionale scheepsontwerpen te genereren. Daarnaast geeft de methodiek inzicht in hoe wijzigingen in de ontwerpeisen het uiteindelijke scheepsontwerp zullen beïnvloeden. Variaties in ontwerpeisen die onderzocht zijn betreffen ondermeer de endurance van een fregat alsmede de sensoren en wapens waarmee een dergelijk schip uitgerust is. Bovendien is een verzameling met haalbare fregatontwerpen gebruikt om de toepassing van de selectieaanpak te illustreren. Door de ontwerpen te kiezen die beter zijn op bepaalde aspecten, bijvoorbeeld werkbaarheid in zeegang, kan inderdaad inzichtelijk worden gemaakt hoe dergelijke keuzes het uiteindelijke scheepsontwerp beïnvloeden.

Samengevat kan gesteld worden dat de nieuwe ontwerpmethodiek, zoals gepresenteerd in dit proefschrift, het mogelijk maakt om snel en vroegtijdig een grote verzameling aan haalbare, driedimensionale scheepsontwerpen te genereren. Daarbij stelt methodiek de ontwerper in staat om al deze ontwerpen te bestuderen en met dit inzicht uiteindelijke de meest interessante scheepsontwerpen te selecteren.

De verwachting is dat met de mogelijkheden die deze nieuwe methodiek biedt, de ontwerper beter in staat om vroegtijdig interessante combinaties van systemen, indeling en prestaties te identificeren, hetgeen uiteindelijk moet leiden tot het ontwerp van succesvollere en winstgevendere werkschepen.

Publications related to this dissertation

Journal publications

1. B J van Oers, M Th van Hees, D Stapersma, J J Hopman, *'Combining a Knowledge System with Computer-Aided Design'*, Ship Technology Research, Vol 55, pp. 51-59, 2008

2. B J van Oers, D Stapersma, J J Hopman, *'An Optimisation-Based Space Allocation Routine for the Generation of Feasible Ship Designs'*, Ship Technology Research, Vol 56, pp. 31-48, 2009

Conference publications

1. B J van Oers and M Th van Hees, *'Combining a Knowledge System with Computer-Aided Design'*, Fifth International Conference on Computer Applications and Information Technology in the Maritime Industries, Leiden, The Netherlands, May 2006

2. B J van Oers and D Stapersma. *'Applying first-principle tools in naval ship design'*, In Proceedings of Warship 2006: Future Surface Warships, London, United Kingdom, June 2006, Royal Institution of Naval Architects.

3. B J van Oers, D Stapersma, J J Hopman, *'Improvements of a knowledge-based CAD system for conceptual ship design'*, Sixth Int. Conf. Computer Applications and Information Technology in the Maritime Industries, Cortona, April 2007

4. B J van Oers, D Stapersma, and J J Hopman, *'Development and implementation of an optimisation-based space allocation routine for the generation of feasible concept designs'*, Sixth Int. Conf. Computer Applications and Information Technology in the Maritime Industries, Cortona, April 2007

5. W H de Bruijn and B J van Oers, *'Submarines for the future, a new design model'*, Ninth International Naval Engineering Conference (INEC 2008), Hamburg, Germany, April 2008, The Institute of Marine Engineering, Science and Technology

6. B J van Oers, D Stapersma, and J J Hopman, *'Issues when selecting naval ship configurations from a Pareto-optimal set'*, Twelfth AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia, Canada, September 2008, American Institute of Aeronautics and Astronautics

7. T W DeNucci, J J Hopman, B J van Oers, *'Capturing Trade-Off Rationale during Design Selection'*, Eight Int. Conf. Computer Applications and Information Technology in the Maritime Industries, Budapest, April 2009

8. T W DeNucci, J J Hopman, B J van Oers, *'Prototype Development of a Ship Rationale Capture Tool'*, 16th International Conference of Ship and Shipping Research, Messina, Italy, November 2009

9. B J van Oers, D Stapersma, and J J Hopman, *'A 3D packing Approach for the Early Stage Configuration Design of Ships*, Ninth Int. Conf. Computer Applications and Information Technology in the Maritime Industries, Gubbio, Italy, April 2010

10. K D Wagner, A Wassink, B J van Oers, J J Hopman, *'Modeling Complex Vessels for Use in a 3D Packing Approach: An Application to Deepwater Drilling Vessel Design*, Ninth Int. Conf. Computer Applications and Information Technology in the Maritime Industries, Gubbio, Italy, April 2010

11. B J van Oers, D Stapersma, and J J Hopman, *'A 3D packing Approach for the Early Stage Configuration Design of Ships*, Tenth International Naval Engineering Conference (INEC 2010), Portsmouth, United Kingdom, May 2010, The Institute of Marine Engineering, Science and Technology

12. K D Wagner, B J van Oers, A Wassink, J J Hopman, *Practical Application of Early Stage Ship Configuration Optimization: Deepwater Drillship Design*, Practical Design of Ship and Offshore Structures Conference (PRADS), Rio de Janeiro, Brazil, September 2010

13. B J van Oers. *Designing the process and tools to design affordable warships*, NATO-RTO-MP AVT-173 Workshop on Virtual Prototyping of Affordable Military Vehicles Using Advanced MDO, Sofia, Bulgaria, 2011.

Acknowledgements

The research presented in this thesis was funded by the Royal Netherlands Naval College (now part of the Netherlands Defence Academy) and carried out at Delft University of Technology.

The Defence Materiel Organisation in The Hague generously offered support in the form of access to design data and, most importantly, a second place to work, which allowed me to perform my research among practising naval architects and marine engineers.

Delft University of Technology and the Defence Materiel Organisation also helped by allowing me to complete this dissertation during working hours, while further contributions came from GustoMSC, NEVESBU and MARIN; the latter organisations provided software, helped supervise MSc. students and shared design data.

I would like to thank all these organisations for their support. Without it, this project would not have been possible.

In addition, I have benefitted from the support and interest from a group of talented naval architects, engineers, academics and students. Not enough space is available to thank everyone by name, but I would like to mention and thank a number of people whose efforts warrant an honourable mention.

The doctoral committee, for your time and effort spent reviewing my thesis. Your comments, suggestions and critique were appreciated. Also, my thanks to Jeantine Schuurmans for reviewing this manuscript and removing a large number of spelling errors.

Dr. ir. Philipp Wolff and dr. ir. Martin van Hees for their inspiring lectures on ship design methodology and ship design tools at Delft University of Technology. Your efforts sparked my interest in this field of research.

Prof. ir. Hans Klein Woud, for supporting me to pursue my own PhD proposal and for enlisting the support of the Royal Netherlands Navy. Prof. dr. ir. Tom van Terwisga and dr. ir. Robert van de Ketterij, for their time and effort spent on supervising me during the first part of the project, and for helping me to get started in a PhD-project.

Kees van Roosmalen at NEVESBU and Alain Wassink and Bas Goris at GustoMSC for their interest and support and, in particular, for allowing me to test results of my research in collaboration with industry. Your support for, respectively, Wendy de Bruijn and Karel Wagner, was very valuable.

The naval architecture group at University College London in general and prof. David Andrews, dr. Richard Pawling and dr. Tim McDonald in particular. The discussions on early stage warship design as well as on life as a PhD student were and still are highly appreciated.

Also, my colleagues at Delft University of Technology, in particular dr. ir. Hugo Grimmelius, ir. Peter de Vos and ir. Elena Moredo, for the company over the past years and for the many memorable conferences we attended together.

Ir. Jaap Huisman, KTZT ir. Hans van Norden, ir. Kees van Es and my other colleagues at the Defence Materiel Organisation in The Hague for their interest and company over the past years. Special thanks to both ir. Robin Brouwer and ing. Marcel Visser for your friendship, support, interest in ship design research and your positive nature in general.

Ferdi Drosten, Martijn van Diessen, Wendy de Bruijn, Gideon van Ingen and Jos Govaerts, for your contributions to the results presented in this dissertation. Thanks again for the effort each of you made in your graduation projects for, respectively, the Royal Netherlands Naval College and Delft University of Technology.

LCDR Thomas DeNucci (USCG) for his willingness and ability to build an entire PhD project on knowledge-capturing in ship design from the single observation that naval architects like critiquing designs created by the approach presented in this thesis. Tom, I appreciated your company the past years and hope you and your family liked living in The Netherlands.

Karel Wagner, for doing a difficult but brilliant job in applying the design approach from this thesis for the very first time (on deep water drillship design). It was a pleasure and honour working with you; you will -most likely- be the only MSc. student I have ever supervised who went on to become the ‘best graduate’ at the Faculty of Mechanical, Maritime and Materials Engineering.

Prof. ir. Hans Hopman for your support, our discussions on ship design and helping me to finish this thesis, as well as your willingness to share your broad experience with warship design at the Netherlands Ministry of Defence.

Prof. ir. Douwe Stapersma, for your willingness to support and fund my PhD project, but above all for training me as a researcher in a very thorough, integer and brutally honest way. Douwe, your ability as a marine engineer to look beyond boundaries and -most importantly- beyond tradition, has changed naval architecture more than many a ship designer.

My parents and my brother, for putting up with a son / brother who paid infrequent visits and who was always busy the past years.

Last, but most certainly not least, Léonie and Isolde, without whose love, patience, understanding and support this project would not have started, would not have progressed and, most certainly, would not have been finished.

Curriculum Vitae

Bart van Oers was born on July 8, 1979 in Breda, The Netherlands. He attended the Atheneum (secondary school) at the Mencia de Mendoza Lyceum in Breda from 1992 to 1997.

From 1997 to 2005 Bart studied Marine Technology at Delft University of Technology. He specialised in Ship Hydrodynamics and, for his MSc. research project, investigated the viscous flow around a large crude oil tanker sailing at a drift angle at the Maritime Research Institute Netherlands. He graduated in April 2005, obtaining a MSc. degree with Honours.

During his time as a student in Delft, Bart participated in the organising committee of a study tour to Singapore, Hong Kong and mainland China in 2000. Also, he worked on a three-month project for the Royal Netherlands Navy in 2004 to develop a prototype knowledge-based CAD system for early stage ship design. Furthermore, Bart wrote the PhD research proposal that formed the basis of the results presented in this dissertation. This proposal was funded by the Royal Netherlands Naval College (now part of the Netherlands Defence Academy).

Bart started to work as a PhD student Delft University in April 2005 to investigate novel approaches to improve the early stage design of complex ships. Part of the results of this effort are presented in this dissertation. His work in Delft also involved teaching a course on ship design methodology, as well as supervising the graduation projects of students (both from Delft University and the Netherlands Defence Academy).

After finishing his PhD research in 2009, but while still writing this dissertation, Bart took up a position as a Senior Naval Architect at the Defence Materiel Organisation in The Hague in December 2009. His job entailed further developing early stage ship design tools and assisting in their implementation in the warship procurement process. This dissertation was finished part-time, next to his day-to-day job and family responsibilities. In November 2011 Bart became team leader platform and is responsible for the naval architectural design of future Royal Netherlands Navy warships during the early stages of procurement.

Bart authored and co-authored numerous conference papers as well as two journal articles. One of the former won the Institute of Marine Engineering, Science and Technology's Sir Donald Gosling Award in 2008 (co-authored with Wendy de Bruijn). Lastly, the research presented in this dissertation was awarded with the 2011 edition of the 'Timmersprijs' for the best Dutch ship design process innovation.

Bart lives with his wive Léonie and their daughter Isolde in The Hague, The Netherlands. They expect a second child in January 2012.