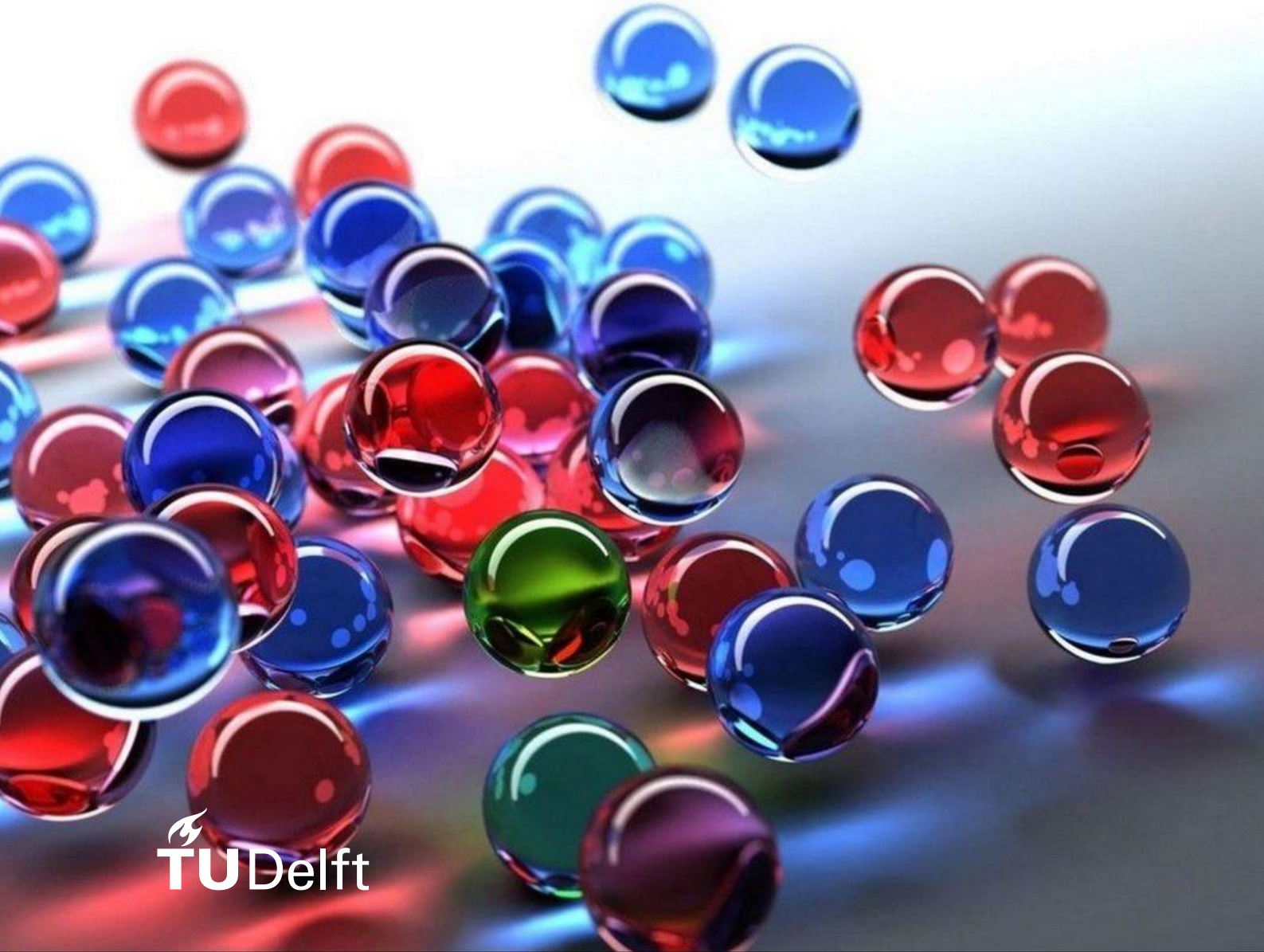
ELUCIDATING FAMILIES OF SHIP DESIGNS USING CLUSTERING ALGORITHMS

T.J.M. Jaspers



TU Delft

Elucidating Families of Ship Designs using Clustering Algorithms

By

T.J.M. Jaspers

in partial fulfilment of the requirements for the degree of

Master of Science
in Marine Technology

at the Delft University of Technology,
to be defended publicly on Monday July 10, 2017 at 9:30 AM.

Supervisor:	Prof. ir. J. J. Hopman	
Thesis committee:	Dr. A. A. Kana,	TU Delft
	Dr. S. Schreier,	TU Delft
	Dr. D. M. J. Tax,	TU Delft

SDPO graduation number: SDPO.17.016.m
An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

Already when I was a little boy, my biggest dream was to become an inventor someday. My role model in life was Gyro Gearloose (in Dutch: Willie Wortel), and every night I dreamt about new creations that I would build. Although I was raised inlands, and had never sailed a ship, one often recurring creation was "the perfect boat". I remember often telling my parents about it, and at the age of 15, I even made a bet with a friend that I would build a boat before my 18th birthday. Being more a thinker than a doer, I of course never did it… The big problem was that I could never wrap my head around what to optimize for: speed, comfort or maybe the ease of building it?

Moving on during my bachelor study's in physics, the fascination for the concept of the perfect boat continued to exist, which motivated me to pursue a master's in ship design. Being used to solving problems analytically, and thus optimal, I was amazed about the often manual and sometimes hand-waving techniques still used to design these massive multimillion-dollar structures. It took a while before it became clear to me how complex the design of a ship really is, realizing that finding only one feasible floating design solution is already a piece of art[1]. Nonetheless the desire to work on a method, that at least tries to design ships more optimal, remained. The direction for this master thesis then became instantly clear when I encountered the TU Delft Packing Approach, a tool that automatically generates thousands of ship designs using a genetic algorithm.

The professor in charge, prof. J.J. Hopman, then told me that while analyzing the results from the packing approach, it seemed that some designs look a lot like each other: "There might even be designs in the dataset which are totally the same, except the bridge is just shifted one meter." It therefore became my job to investigate to what extend this was true, and if maybe the set of thousands of designs could be reduced to just a handful of different families of designs.

Although this subject seemed to be valid, one of the main struggles during this thesis was defining exactly the need for this research. The puzzle pieces started fitting together when Koen Droste came up with an article stating that fundamentally, data first has to be structured in order to get to knowledge. Thus the reason why you would look whether designs can be classified into families, is the same reason why you would just plot two variables, and see how they are correlated; you just structure the data in various ways, and see whether new knowledge appears.

This motivates a special thanks to my office mates, Koen Droste and Agnieta Habben Jansen, for giving such insights and discussing related subjects (often while drinking a nice cup of coffee). I furthermore gratefully thank my daily supervisor, Austin Kana, who has not only been amazingly kind and helpful, but mainly distinguished himself by being sincerely interested in the progress. The same holds for Bijan Ranjbarsahraei, who additionally supported me by sharing his wisdom on the various techniques and methods from the field of data science in regular meetings, and prof. Hans Hopman, for pointing me in the right direction and discussing the bigger picture, even with an overall lack of time.

[1] Probably every naval architect in training experiences this moment.

Last but definitely not least, I want to thank my family and girlfriend. Thanks sister, for your remaining interest in the normal human being, despite your incredible set of brains, thus helping me a lot with going through processes like these. Thanks mom and dad for the immense support I'm getting, every time I need some. And thank you, Simone, for talking about the important things in life in Limburgs, making phone calls through random objects, discussing whether to eat a sandwich with a croquette, and for dropping nearly every eatable item on your clothing. Thus for making me laugh.

T.J.M. Jaspers
Delft, July 2017

Contents

Preface 3
Contents 7
List of Figures 9
List of Tables 11
Glossary 12
Abstract (Summary) 13
1. Background 14
1.1 DIKW pyramid 14
1.2. Concept exploration 15
1.3 The packing approach 16
1.4 Families of designs 19
1.4.1 Relating families 19
1.4.2 Hypotheses generation 21
1.4.3 Data reduction 22
1.5. Scope of this thesis 22
1.6. Research objective 25
2. Initial packing results investigation 26
2.1. Design comparison 26
2.1.1 Compare physical attributes 27
2.1.2 Chromosomal comparison 28
2.2. Set comparison 29
3. Method 35
3.1 Feature selection/creation 36
3.2 Proximity measure 37
3.3. Dimensionality reduction 38
3.3.1 Principal Component Analysis 38
3.4. Clustering 39
3.4.1 K-means 40
3.5. Validation 42
3.5.1 within cluster sum of squares (WCSS) 42
3.5.2 Dunn-index 43
3.6 Interpretation 43
4. Test Case: Mine-countermeasures vessel 44
4.1. Survivability of machine systems 44
4.1.1. Iteration 1 45
4.1.2. Iteration 2 47
4.1.3. Iteration 3 49
4.2. Families based on layout 50
4.2.1. Design selection based on designer rationale 51
4.2.2 Apply method 52
4.3 Chromosome analysis 57
4.3.1. Iteration 1 57

4.3.2 Iteration 2 58
5. Discussion & Future work 60
5.1 Continue the assessment of survivability of machine systems 60
5.2 Lacking diversity in Packing 60
5.3 Reduced layout space 61
6. Conclusion 62
7. Contributions 63
Bibliography 64
Appendix A: COMPIT paper 66

List of Figures

Figure 1: The DIKW pyramid as presented in *Rowley (2006).* 15
Figure 2: Illustration of the main purpose of concept exploration; investigate the relation between design and performance space *Duchateau (2016)* 16
Figure 3: Illustration of the packing approach *Duchateau (2016)* 17
Figure 4: Visualization method of packing data as described by *Duchateau (2016)*. Various features are selected for plotting in matrix plots and interactively constraints can be added. . 18
Figure 5: In this artificial dataset no clusters are detected by looking at 2d plots of (a), (b) and (c), whereas looking in 3d does reveal two clusters (d). 19
Figure 6: Illustration of how families in the design space could be related to certain parts of the performance space 20
Figure 7: Various designs plotted in performance space, divided into families based on luxury level, *Droste (2016)* 20
Figure 8: Illustration of hypothesis generation 21
Figure 9: Illustration of data reduction. 22
Figure 10: The design spiral as presented in *Evans (1959)*. Packing only calculates the outer circle. 23
Figure 11: Systems engineering V-diagram as used in *Hopman (2017)*, with scope of this research included. 24
Figure 12: Representation of the framework that decomposes the information needed for a system, as presented in *Brefort et all (2017)* 24
Figure 13: Cost versus OPEX plot of all designs resulting from the packing algorithm as used in *Droste (2016)*, including the selection constraints for the six designs in Figure 14. 26
Figure 14: Side view representation of the six selected cruise ship designs 28
Figure 15: Example of a MCMV design from *Duchateau (2016)* 30
Figure 16: Six histograms of the gene values for the cruise ship 32
Figure 17: Six histograms of the gene values from the MCMV 33
Figure 18: Logarithmic histogram of the x-position of the main gun. Three families emerge: no gun, gun on the aft or at the bow. 34
Figure 19: Four MCMV-designs resulting from the packing approach, where various differences and similarities are pointed out. 36
Figure 20: Six step clustering method visualized with LEGO blocks. Every block stack is a new method on its own. 36
Figure 21: Illustration in 2D how PCA rotates the data. It makes it as "flat" as possible. 39
Figure 22: PCA applied to the data used in Figure 5. The first pc lies in the direction that reveals the clusters. 39
Figure 23: Hierarchical chart showing a coarse division of the different type of clustering algorithms. 40
Figure 24: Illustration of how k-means converges on a 2D artificial dataset for k=3 41
Figure 25: Example of a knee in plotting the WCSS versus k at k=5. This indicates that the data contains 5 valid clusters. 42
Figure 26: MCMV objects related to the survivability of machine systems and their interrelations 45
Figure 27: Clustering method used in this section 45
Figure 28: PCA result: Explained variance vs. the number of pc's 46
Figure 29: Biplot showing the factors of the first two pc's 46
Figure 30: Machine systems data projected on the first two pc's, and colored regarding gun position 47
Figure 31: Machine systems data projected on the first two pc's, and colored regarding tank top height. 47

Figure 32: The WCSS for various values of k resulting from k-means, showing a knee at k=3. 48
Figure 33: Projection of the data on the first two pc's and colored by the groups resulting from k-means for k=1 to k=6 48
Figure 34: Projection of the data on its first two pc's. 49
Figure 35: The WCSS for various values of k, resulting from k-means. 49
Figure 36: Illustration of existence of families due to connections between the compartments. 50
Figure 37: The best ships regarding the metrics defined in Table 5: High rank officers near the bridge (a), seasickness bridge (b), accom grouped (c), and noise (d). 52
Figure 38: LEGO stack corresponding to the method applied in this section..... 53
Figure 39: Explained variance vs. the number of pc's used for the MCMV dataset including designer rationale. 54
Figure 40: MCMV dataset including designer rationale plotted regarding its first 2 pc's..... 54
Figure 41: Data plotted in first two pc's clustered by the result of k-means for k=3 55
Figure 42: Validation of the results from k-means using the Dunn index 55
Figure 43: Histogram of Dunn index values for 1000 samples when the dataset is split into two groups by a random hyperplane. 55
Figure 44: Data plotted in first two pc's clustered by the run number in which the data was generated. 56
Figure 45: Chromosome data projected on its first three pc's, where colors correspond to the various run numbers of the packing approach..... 57
Figure 46: Chromosome data projected on its first three pc's, where cluster 8 resulting from the k-means algorithm is highlighted in blue. 59

List of Tables

Table 1: Table of the main properties of the six selected cruise ship designs 27
Table 2: Comparison of the designs by normalized city block distance 28
Table 3: Comparison of the designs by Hamming similarity 29
Table 4: Composition of the chromosome as defined for the MCMV model 31
Table 5: Applicable designer rationale for the MCMV from *Denucci (2012)*, including the metrics representing the rationale. Every metric should be minimized. 51
Table 6: Comparison of the clusters resulting from k-means with the various runs. 58
Table 7: Calculation of the number of possible families 60

Glossary

Term	Definition
Boa, Loa	Beam/Length overall
c_b	Block coefficient, the ratio of the volume of the submerged part of the ship, to the volume of the tightest box around this submerged part.
CoG	Centre of Gravity
dcd	Damage control deck, the lowest deck with access through the transverse bulkheads.
displ	Displacement
GM, GMt	Distance between CoG and the transverse metacentre height (positive if M is above G), which is a metric for initial stability of a ship.
IECEM	Interactive Evolutionary Concept Exploration Method, method proposed by *Duchateau (2016)* for enhancing concept exploration, by analysing results from the packing approach where numerical and architectural constraints could interactively be added.
k-means	A clustering algorithms that divides a dataset into k clusters.
MCMV	Mine-countermeasures vessel
NSGA II	A multi-objective genetic algorithm
Packing density	The ratio of the volume of the objects in a design, to the volume of the total design. Thus a high packing density indicates little empty space.
pc	Principal component, a basis vector resulting from PCA
PCA	Principal Component Analysis, a multi-variate data analysis method that rotates data, such that the first principal components align with directions of high variance.
RCT	Rationale Capturing Tool, the tool developed by *DeNucci (2012)* that captures and stores designer rationale from naval architects.
SOM	Self-organizing maps, a dimensionality reduction method that maps mostly a 2D or 3D grid to higher dimensional dataset, which can be used to visualize the underlying structure.
t-SNE	t-distributed Stochastic Neighbour Embedding, a dimensionality reduction method that mimics the structure of data, which is visualized in a 2D or 3D plot.
WCSS	Within Cluster Sum of Squares, the sum of Euclidean distances between every data point and its corresponding cluster centre. This is often used to validate clusters resulting from the k-means algorithm.

Abstract (Summary)

The past decade the packing approach was developed at the TU Delft. This tool automatically generates tens of thousands of ship designs in order to fully explore the design space. This is opposed to the traditional way of ship design, where just a limited number of designs can be elaborated. The result is a set of ship designs that can be used both to obtain initial sizing parameters of the ship, and to relate the impact of design decisions on performance characteristics.

It is questioned whether resulting sets of designs really contain tens of thousands of different ships, or just a couple of really different ships that have only minor variations. It is thus questioned whether such a set of ship designs can be divided into families. This is investigated in this thesis.

In order to attack this problem in a generic fashion, the ship designs were approached from a numerical perspective. It was found that in the field of data science, clustering algorithms exist which are devoted to find clustering structures in data. Therefore these techniques, such as PCA and k-means, were applied to data at hand. In a test case this method is applied to divide the set of designs from a mine countermeasures vessel into families.

First it was questioned whether families could be used in order to assess the survivability of machine systems of these ships. The resulting families matched the families regarding the position of the gun, which were already known upfront. Although these families are not sufficient to fully assess the survivability of these ships, this analysis showed that most probably no other structure is present, stimulating more straightforward definitions of families.

In a second test case the same MCMV was divided into families regarding the layout of the designs. It then appeared that the dataset at hand was built by ten distinct runs of the packing approach which were combined to this one total set. The method showed that it could be pointed out, just by looking at the designs, which design was generated in which run. This showed that the dataset might not be as diverse as it seems. Since it is the core idea for the packing approach to explore a big part of the design space, thus generating a diverse set of ship designs, this motivates new research to tackle this issue.

1. Background

Roughly ten years ago, B.J. van Oers started his PhD, and developed the packing approach *van Oers (2011)*. The goal of this tool was to be able to automatically generate a vast number of ship designs, and thus explore a big part of the design space. When this objective was completed, and datasets were generated with thousands of ship designs, the next problem was how to analyze this data. This problem was attacked during the PhD thesis of *Duchateau (2016)*, where he developed a method for plotting variables, and interactively adding constraints to these plots. This method revealed a lot of information, but also generated new questions, since sometimes families of designs appeared in his plots. It was therefore questioned whether there were more of these structures present, and how they could be found. This led to initiating this MSc thesis project[2].

In sections 1.2 to 1.5 the fundamental concepts of this thesis are elaborated, which are:

1. The *concept exploration* phase within the early stage ship design, since this is the phase of the design process this thesis is operating in.
2. The *packing approach*, which is the tool developed at the TU Delft that generates the results used in this thesis.
3. The concept of *families of designs*, and why finding these families could be helpful.

Finally in section 1.5 is described how this thesis fits in a broader perspective of ship design, and in section 1.6 the research objective is defined. But before discussing these fundamental concepts, this chapter is initiated by stating its basic axiom in section 1.1: the Data-Information-Knowledge-Wisdom pyramid.

1.1 DIKW pyramid

Before going into the design of ships, this chapter starts on a more philosophical level by introducing the Data-Information-Knowledge-Wisdom pyramid (DIKW pyramid), which was first described by *Ackoff (1989)*, and is displayed in Figure 1. Although often assumed implicitly, it is believed by the author that this pyramid is the basic axiom for justifying the need of this thesis. It shows that in order to get to knowledge and finally wisdom, data has to be gathered and converted into information, which in turn has to be interpreted. This corresponds to the structure of this thesis, since data is generated by the packing approach that is tried to be structured into families (information) in order to acquire knowledge and wisdom. The pyramid can be explained as follows[3], using the definitions by *Ackoff (1989)*:

- Data is a collection of symbols. As an example assume having data collected on various ships, including seasickness of their captains.
- Information is data that is processed to be useful. In the example plots can be made from the data, including a plot of the seasickness of the captain versus the position of the bridge.
- Knowledge is the application of data and information. It answers "how" questions. Answering how seasickness of the captain and position of the bridge are related, the

[2] Parts of this thesis has been published in *Jaspers and Kana (2017)*.
[3] There is still disagreement on the exact shape and definitions, although there seems to be consensus that the definition of each element (except data) uses its predecessor *Rowley (2006)*.

plot shows that when the bridge is closer to the bow, the captains suffer more from seasickness.

- Wisdom is the appreciation of why. When the bridge is closer to the bow, it is further from the ships center of rotation. Following the law of the lever, the bridge, including the captain, will encounter higher accelerations, which causes seasickness.

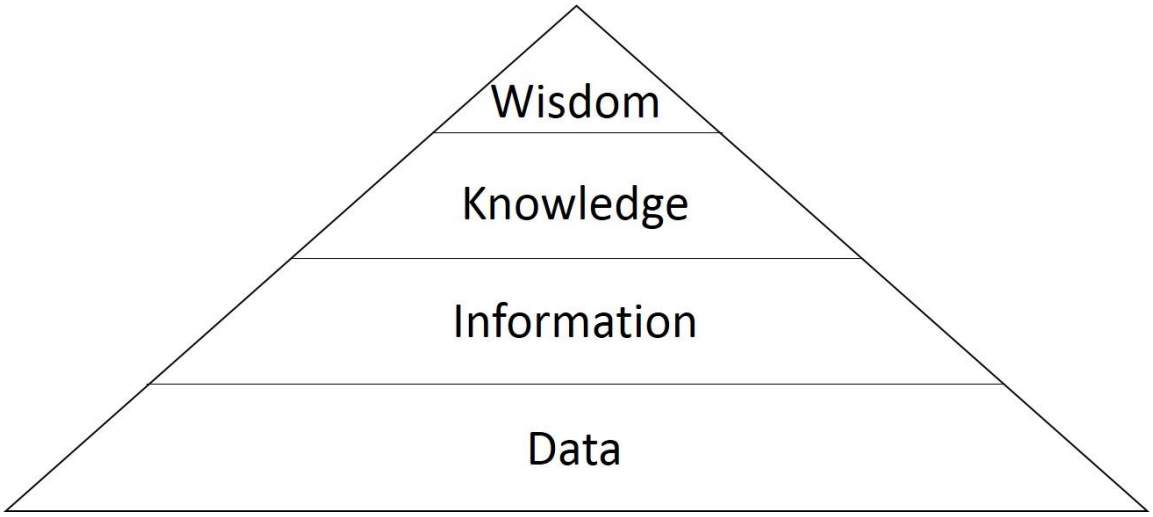


Figure 1: The DIKW pyramid as presented in *Rowley (2006).*

In the jargon of ship design, acquiring knowledge is often seen as the goal, instead of generating wisdom. This is no problem, since the process of converting knowledge to wisdom does not need planning or description. People inherently try to explain why things happen. Thus in the rest of the thesis when the term knowledge appears, assume that wisdom will always result from it.

1.2. Concept exploration

Early stage design is the initial phase in ship design where the balance between the different desired performances of the ship is explored. The result is normally one design that is further elaborated during the subsequent contract design phase. Early stage design is often initiated by performing concept exploration. The main goal of concept exploration is acquiring knowledge about how designs decisions influence performance characteristics *Duchateau (2016)*. This means investigating the relation between design space (spanned by the design variables) and performance space (spanned by the performance attributes), which is illustrated in Figure 2. Acquiring knowledge means running through the DIKW pyramid, therefore, as data source, various concept designs are elaborated. But, as is argued in *Duchateau (2016),* in ship design there are typically many dependencies and interactions between the design variables. In order to map this complex network of interrelations, the design space has to be searched thoroughly.

Another typical property of ship design is the vast number of design options, which results in a very high dimensional design space *Duchateau (2016)*. Examples for variations start on a high level as there are often various functions needed, with multiple distinct system configurations covering the same functional requirements, which in turn consist of multiple possibilities for sub-systems. Additional there are numerous design variations, such as length, beam, positions of all compartments, sizes of the compartments, hull shape, shape of the

superstructure, and so on. Combining the high dimensionality of the design space with the need for a thorough search means that a vast amount of designs has to be elaborated.

During concept exploration of complex ships, new requirements and/or new relations between requirements can be elucidated. This results in an iterative process called requirements elucidation *Andrews (2013)*. On top of that, the traditional method of manually iterating through the design spiral is very time consuming. These aspects cause that in general only a small part of the design space is explored, which leads to an increased probability of converging to a suboptimal design, *Vasudevan (2008)*, *Duchateau (2016)*. In order to explore the design space more extensively, the 'packing approach' was developed at the Delft University of Technology, which automatically generates tens of thousands of coarse feasible ship designs.

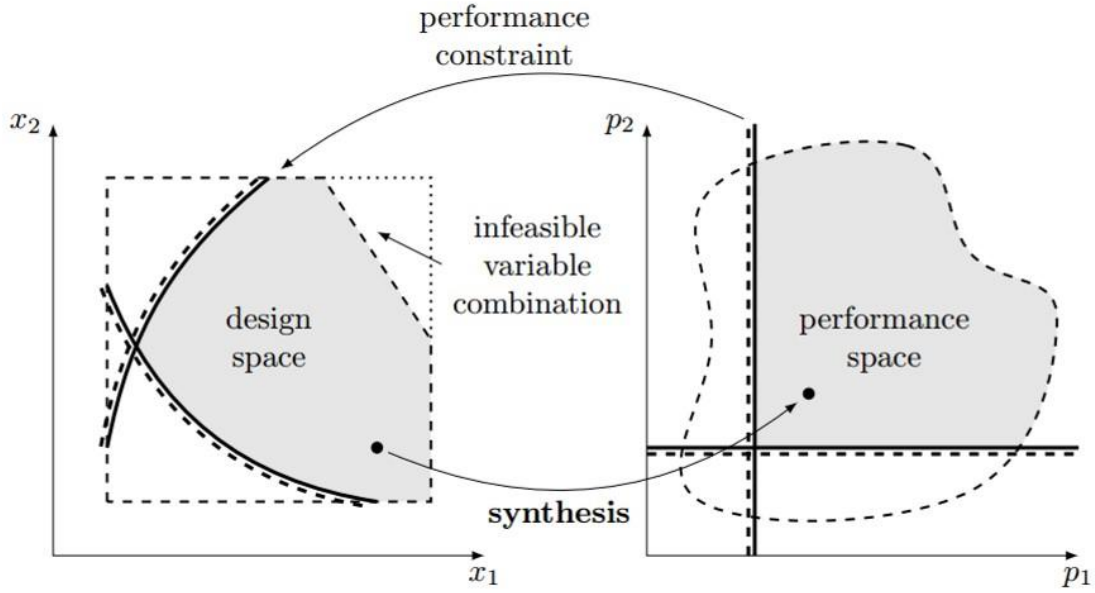


Figure 2: Illustration of the main purpose of concept exploration; investigate the relation between design and performance space *Duchateau (2016)*

1.3 The packing approach

The packing approach is a tool that assists in the concept exploration process. The idea is to automatically generate a vast number of low level of detail feasible ship designs that cover a significant part of the design space. The resulting dataset of ships can then be used to identify design drivers and trade-offs for the particular design, and to deduce initial sizing parameters for starting the design process. The packing approach, as displayed in Figure 3, consists of three steps which are iterated to constitute the desired set of ship designs:

1. The packing algorithm includes the parametric model of the ship. Its input is a chromosome vector of values between zero and one, which is converted to a ship description regarding the ship model. This model consists of the definition of its building blocks and the rules of how these blocks may be packed (packing-rules), which is all defined upfront for a particular design.
2. Then performance measures are calculated for the design. Examples are: packing density, building cost, operating cost, resistance/speed, stability and displacement.

3. These performance measures are fed into a genetic algorithm (NSGA II) allowing the values for the predefined objective function and constraints to be calculated. The default objective is maximizing the packing density[4], thus stimulating the generation of denser designs. If the constraints are met, the design with its performances is stored. These constraints are typically non-negotiable requirements as: sufficient initial stability, all objects are packed, and the initial assumed design draft exceeds the final calculated draft (guaranteeing that the design speed is reached). The genetic algorithm returns a new chromosome, initializing a new iteration. It uses a very high mutation rate, intended to result in a high diversity[5]. This motivates referring to the algorithm as a search algorithm instead of an optimization algorithm.

This process and more details of this approach can be found in *van Oers (2011), van Oers (2012)* and *Duchateau (2016).*

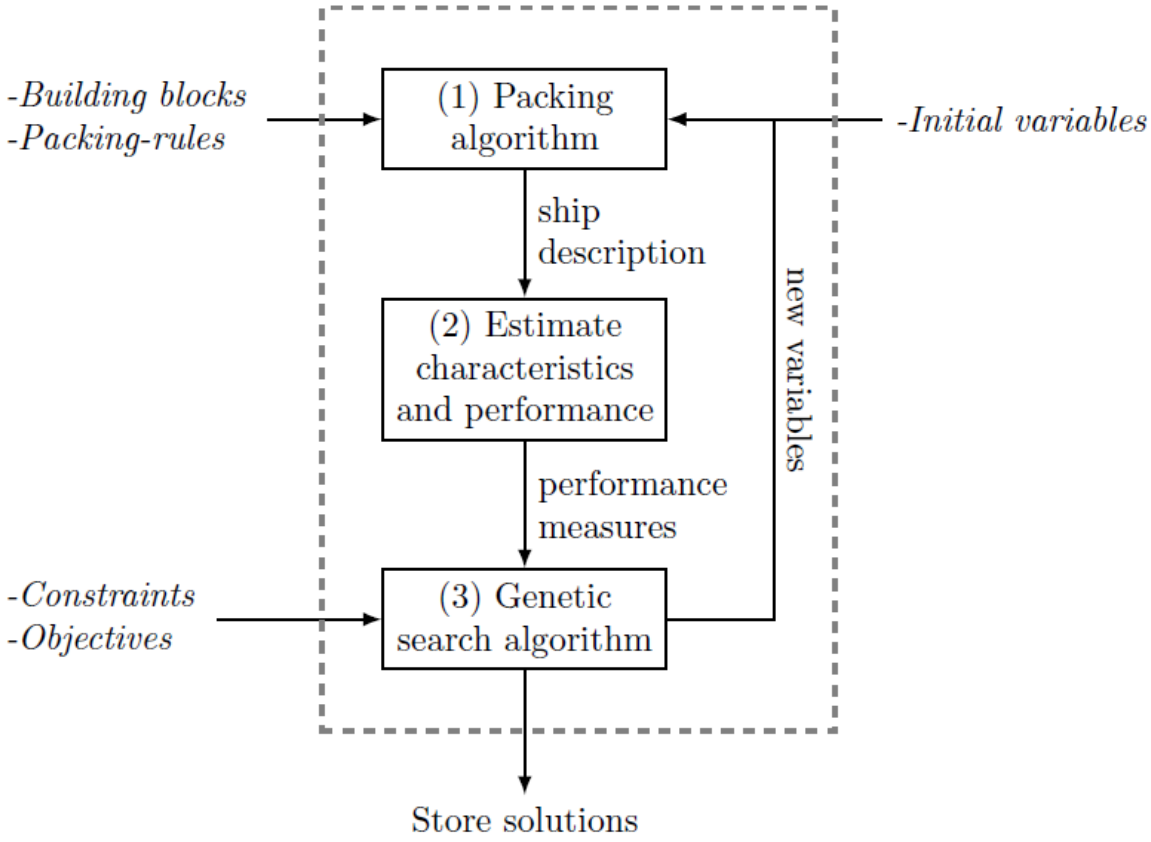


Figure 3: Illustration of the packing approach *Duchateau (2016)*

The resulting data set may consist of tens of thousands of designs, where each design has hundreds of design and performance attributes. Structuring and visualizing this data converts it into information, so that knowledge might be acquired. One visualization method applied to the packing approach is described in the Interactive Evolutionary Concept Exploration Method (IECEM) by *Duchateau (2016)*. He proposed a method of displaying the data in

[4] In practice it minimizes the negative packing density, which is equivalent.
[5] Diversity in this context is different from diversity as used in *Doerry (2015)*, where it is defined as a metric measuring the adaptability of systems in a single design while remaining feasible.

matrix scatter plots, where numerical and architectural constraints could interactively be added. In Figure 4, L, B, GM and packing density are plotted, and the constraint added is that designs have deck 4 as damage control deck (dcd) instead of deck 5. Several results can be deducted from this figure. The bottom left plot shows for instance that the length and packing density are negatively correlated (this is knowledge). The reason is that since a longer design has more space available, it has therefore more empty space to fit the same number of objects (this is wisdom). Furthermore looking at the blue group (where dcd = 4) versus the grey group (where dcd = 5) in the middle plot, a higher dcd results in a lower GM. This is because objects (such as the main gun) should be located above dcd, which raises the center of gravity, and thus lowers the GM.

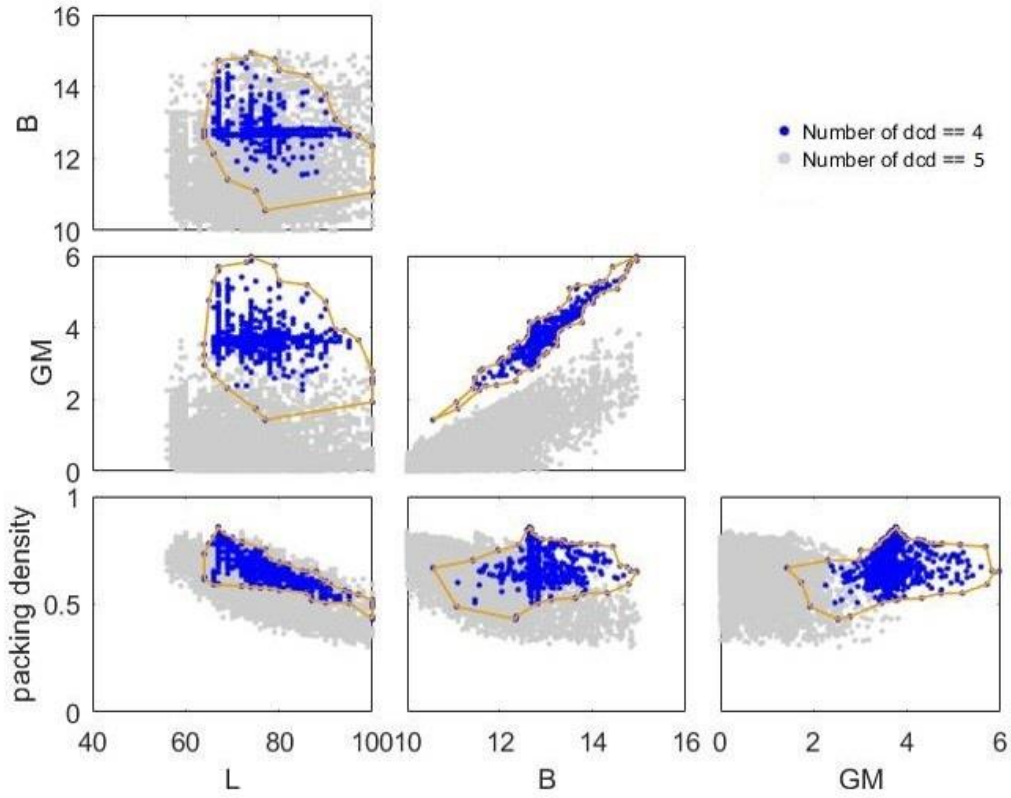


Figure 4: Visualization method of packing data as described by *Duchateau (2016)*. Various features are selected for plotting in matrix plots and interactively constraints can be added.

Although the plots in Figure 4 contain a lot of information, they also raise new questions. Looking for instance at the families present in the middle plot, they are not separated in the plot in the lower left corner. Realizing that these plots all represent the same multi-dimensional data cloud, projected on different 2D subspaces, this then automatically raises the question whether other perspectives might reveal other families. These other families might then be even more different than the families already obtained in these plots. Or more importantly, these families might reveal new knowledge of the data at hand. These questions are the essence for initiating this research project, which is further elaborated in the next section.

1.4 Families of designs

The goal of this thesis is to investigate whether the resulting designs from the packing approach are dividable into families of ship designs. Families of ship designs are defined as being subsets of designs that share a clear similarity within design and performance attributes. Additionally these families should be clearly different when comparing designs between different subsets. The justification of this goal finds its origin in the DIKW pyramid: it's another way (besides the IECEM) of structuring the data, thus converting it to information. This in turn might result in acquiring new knowledge of the relation between design and performance space, which might not have been possible before.

Figure 4 already showed that certain families can be detected using the IECEM. But in order to show that the IECEM is not sufficient to reveal all structure, the artificial dataset in Figure 5 was created. Looking at the data from the 2D plots (Figure 5a-c), there is no special structure present. But rotating the data in 3D space (Figure 5d), reveals that the data actually consists of two distinct families. These are exactly the type of structures that are sought in this thesis. Examples of how the identification of families might result in knowledge are:

1. Relating families: When families in the design space relate to specific parts of the performance space.
2. Hypothesis generation: Explaining why these families appear.
3. Data reduction: When the families hand a structure to the naval architect for systematically analyzing it.

These examples will be elaborated further in the following sections.

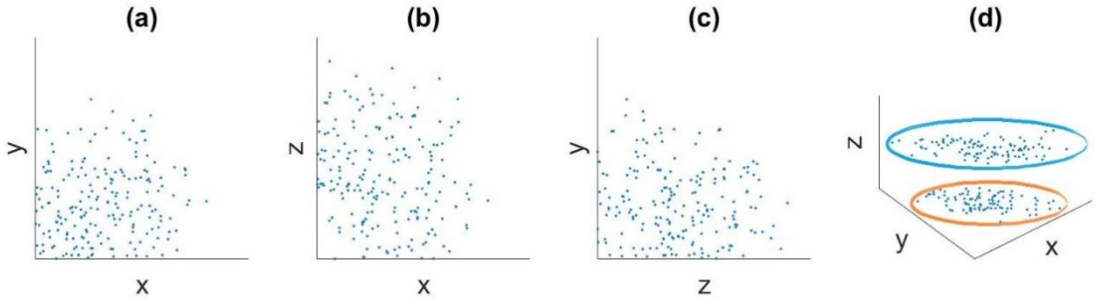


Figure 5: In this artificial dataset no clusters are detected by looking at 2d plots of (a), (b) and (c), whereas looking in 3d does reveal two clusters (d).

1.4.1 Relating families

Finding families of designs could aid in predicting certain performance characteristics. This happens when the families of designs correspond to specific regions of the performance space as is illustrated in Figure 6. The naval architect would then have a better understanding of the relation between design and performance space, thus improving the quality of concept exploration.

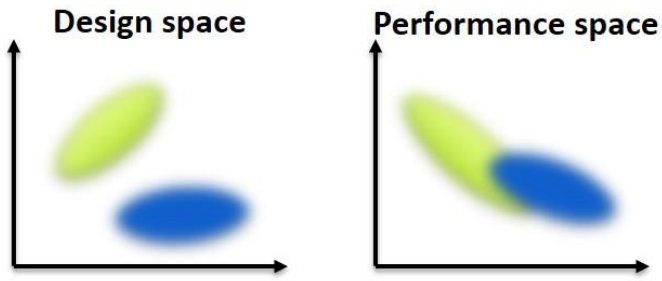


Figure 6: Illustration of how families in the design space could be related to certain parts of the performance space

An example is presented in *Droste (2016)*, where he defined different luxury levels for a cruise ship design. Examining the impact of these families (regarding the design parameter "luxury level") in a performance space is shown in Figure 7. High luxury level causes a jump in both building costs and earning potential, creating two distinct families, clearly showing a tradeoff to be made.

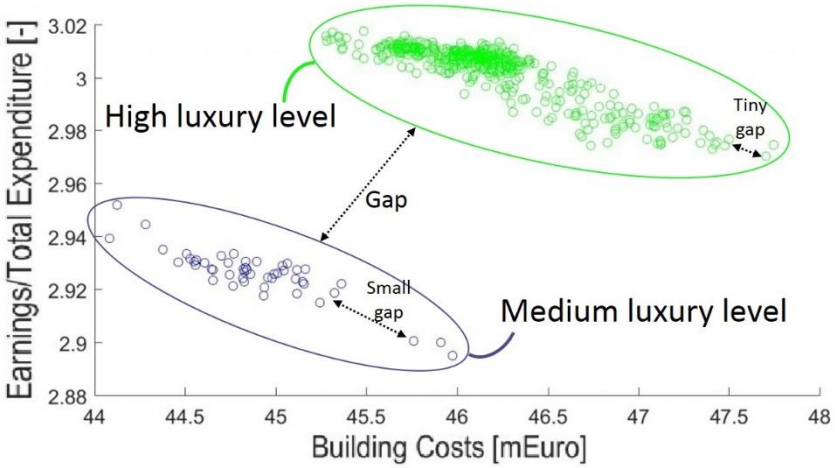


Figure 7: Various designs plotted in performance space, divided into families based on luxury level, *Droste (2016)*

When such a gap appears as in Figure 7, it is either not possible for a design to fit in the gap, or the algorithm failed to find such designs. Which of these two hypotheses to accept, is likely to be motivated by explaining why the gap occurs. For this case the explanation that implementing a higher luxury level causes both higher building costs as earnings seems valid, thus is expected that it is not possible for a design to be in the gap. On the other hand, the small gap within the medium luxury level family could be caused by the algorithm failing to find designs in this region, especially since in the same region for the higher luxury level, this gap is much smaller.

Another example can be found in *Sileryte et all (2016)*, where a parametric varied design set of a swimming stadium is analyzed using multivariate techniques. Various clusters are identified, which relate to particular performance characteristics. For instance several clusters have very poor aesthetical performance, which are thus discarded.

1.4.1.1 Analogy with machine learning

Machine learning is the field of study where algorithms are developed that are able to learn from existing data (referred to as the training set) in order to make predictions *Fox and Guestrin (2015)*. An example is the prediction of house prices from known features such as location, living area, presence of a garden and volume. There is a wide variety of predicting algorithms available, but an often used approach is artificial neural networks. An important step in building a good predictor is feature building, where new features are being built from the existing features which are important to make better predictions. An example is adding the feature "garden area", when the area of the lot and first floor living area are known. Another example is clustering the houses. Whether a house then belongs to a certain clusters (for instance corresponding to our notion of social housing), could be important to predict its price. Applying clustering algorithms for feature building is a commonly used technique *Liu and Motoda (2008)*.

In the analogy, concept exploration is the process of training a natural neural network (the naval architect) to predict performance characteristics from design features. 'Predicting performance characteristics from design features' in this sense thus corresponds to 'understanding the relation between design space and performance space'. Finding families of ship designs in the design space could then reveal new features to allow the naval architect to make better predictions of its performance characteristics. This shows why the identification of families could be useful.

1.4.2 Hypotheses generation

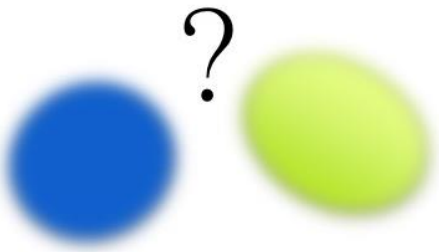


Figure 8: Illustration of hypothesis generation

Finding families of designs automatically leads to the generation of hypotheses about why these families exist. In fact, these hypotheses might elucidate the discontinuous response behavior caused by continuous input variables as discussed by *Duchateau (2016)*. He argues that the continuous longitudinal position of a working deck may cause discontinuous behavior in displacement, due to the restructuring of the top-deck layout around the working deck. He finally concludes that *Duchateau (2016, pp. 10)*: "The challenge lies in identifying when these jumps occur".

The middle plot of the GM value versus beam in Figure 4, which consists of two families serves as an example of hypotheses generation. When these families were detected, it was first hypothesized that the height of the double bottom (which could be either 1m or 1.5m) caused the gap. It was expected that all objects would increase in height, resulting in a higher

center of gravity (CoG), causing a lower GM value. However, this effect was apparently too small to cause the gap. Later it was confirmed, as illustrated by the blue color, that the families are caused by the damage control deck (dcd) being either deck 4 or 5. The reasoning is very similar, since many objects should be above dcd, resulting in an increased CoG when the dcd is increased.

1.4.3 Data reduction

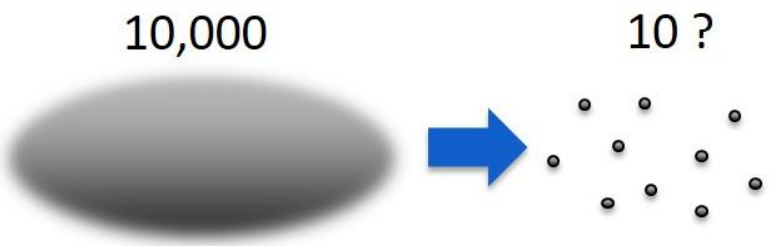


Figure 9: Illustration of data reduction.

As is illustrated in Figure 9 it is questioned whether the tens of thousands of designs resulting from the packing approach are really different, or whether the total set can be represented by only a handful. This might be obtained by dividing the set into families and selecting a representative design for each family. Since the manual analysis of a ship design is time consuming, reducing the data would aid the naval architect in selecting which designs to investigate.

Additionally it might be possible to reduce the data, using data characteristics corresponding to a specific performance characteristic. If families do emerge, an initial estimation of the performance characteristic could then maybe be obtained. This is only the case if the performance is roughly constant for designs within a family. For instance aesthetics is a typical performance characteristic which is hard to quantify, and is therefore often manually investigated. But since aesthetics is mostly depending on the silhouette of the design, reducing the data regarding the silhouette parameters of the design could therefore aid the designer to efficiently quantify such a performance characteristic.

1.5. Scope of this thesis

In this section is discussed how this thesis, and the packing approach in general, fit in the broader perspective of ship design. A widely used and fairly simple illustration of the ship design process is the design spiral, which was first proposed by *Evans (1959)* as depicted in Figure 10. Advantages of the design spiral are the illustrations of both the inherent iterative nature of ship design, and the dependencies of each individual step on its predecessors *Andrews (1998)*. Downsides are however that convergence is not guaranteed *Hopman (2017)*, the order of calculation of its elements can vary per ship type and iteration *Andrews (1998)*, and it only describes the technical aspect of the design, thus omitting the holistic perspective.

The packing approach can be viewed as repeatedly running through the outer circle of the design spiral, since it defines or calculates every aspect of a particular design just once. Assuring the convergence of the spiral during next iterations is obtained by implementing

various constraints. An example is the draft T (which is an input value), and is amongst others used to calculate the necessary propulsion power. At the end, when weight estimations are completed, a constraint is verified whether the new draft T' is smaller or equal to the initial design draft T. This ensures that the propulsion plant is at least big enough.

Furthermore, the order of calculating each aspect is different from the order depicted in Figure 10. The initial fixed values are the hull shape and size (L, B, T), followed by the internal compartment division (deck and bulkhead[6] positions). Next the required size for the propulsion room (and potentially other objects) is determined, which is followed by the packing process, meaning that the positions of all objects are determined. Finally performances such as cost, weight and GM are calculated and corresponding constraints are evaluated.

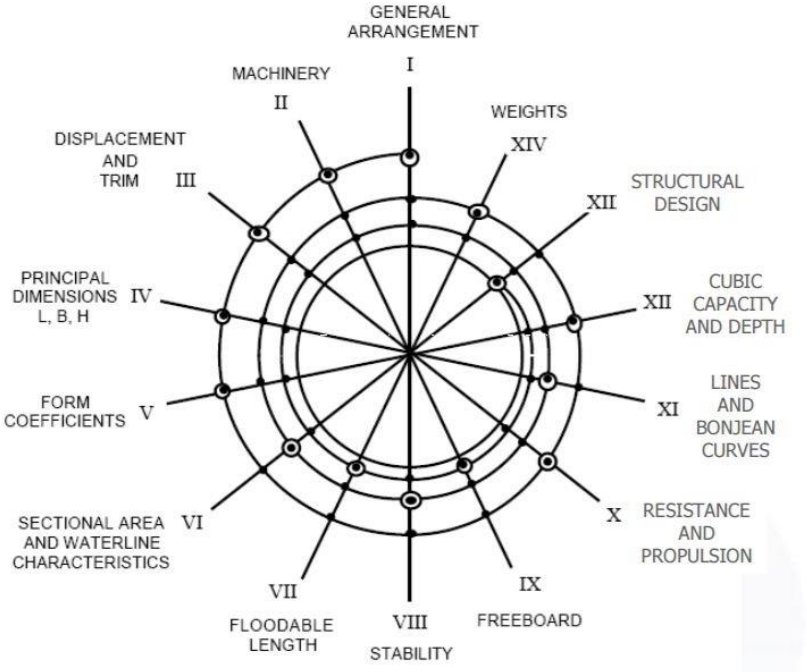


Figure 10: The design spiral as presented in *Evans (1959)*. Packing only calculates the outer circle.

The scope of this research however is harder to grasp using the design spiral, since it is 'the' model for designing a single design (i.e. point-based design), while this research is based on a set of designs, which fits better in the philosophy of set-based design *Singer et all (2009)*. On the other hand, in the V-diagram depicted in Figure 11 the scope of this research can be visualized. An advantage of this diagram, resulting from the field of systems engineering, is that it shows the design system being part of a bigger system (i.e. the holistic view). In this representation, the packing approach itself deals with integration of the physical architecture. This thesis however (just as the IECEM), engages on the data resulting from the packing approach, aiming to both understand the structure of the design space itself as to relate design attributes to performance characteristic, thus expanding the scope as indicated in Figure 11.

[6] An exception is that bulkheads can sometimes be replaced in order to increase the probability of packing large systems such as the propulsion plant.

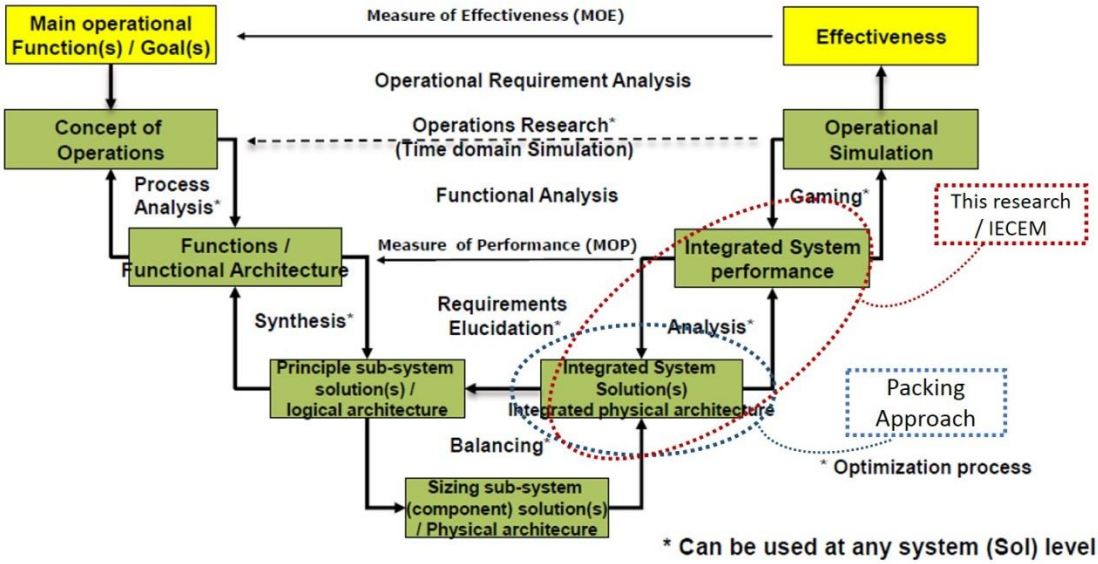


Figure 11: Systems engineering V-diagram as used in *Hopman (2017)*, with scope of this research included.

Finally, Figure 12 represents a decomposition of information needed for the design of a ship, as presented in *Brefort et all (2017)*. The three main classes of information are identified to be the physical, logical and operational architectures which represent respectively the spatial information, functional relationships between components, and temporal behavior characteristics. Within this framework, the research presented in this thesis focusses on the physical architecture of the designs. This class can in turn be divided into information on the overall ship configuration, and information on the components of a certain distributed system, where this research focusses on the former.

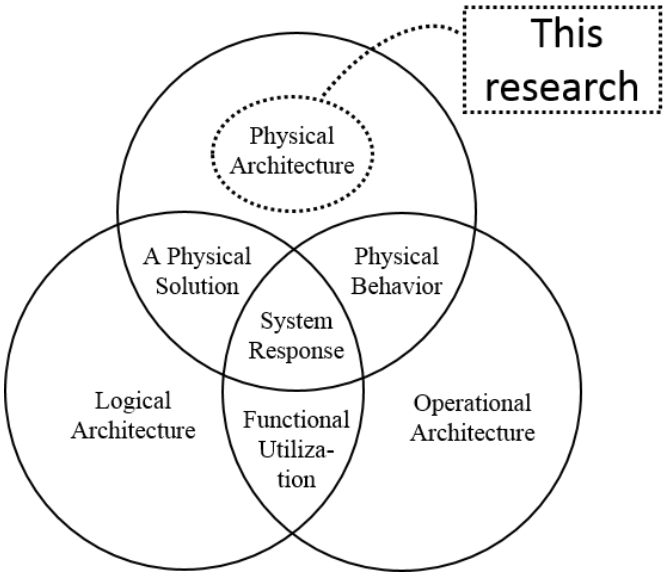


Figure 12: Representation of the framework that decomposes the information needed for a system, as presented in *Brefort et all (2017)*

1.6. Research objective

This thesis aims to expand the amount of information extracted from the data resulting from the packing approach in order to enhance the concept exploration phase in ship design. This is obtained by dividing the resulting design sets into families of ship designs. Therefore the research objective reads:

Elucidate families of ship designs within the design set resulting from the packing approach, leading to new knowledge of the data at hand.

A first step in approaching this objective is investigating which designs are resulting from the packing approach, how these designs can be compared, and which differences between various existing sets resulting from the packing approach exist. This is elaborated in chapter 2. Next, in chapter 3 the method for finding these families is discussed, which is in turn applied to various test cases in chapter 4. Finally, the results feed into the discussion, conclusion and contributions in respectively sections 5-7.

2. Initial packing results investigation

In order to get a hang of the data sets at hand, some initial investigations are performed in this chapter. Since families are defined on the basis of comparison between designs, it is first investigated what it means for designs to be either different or similar. Second a comparison is made between the available design sets, revealing certain characteristics of these sets, and enabling making a decision which data set is best for a test case.

2.1. Design comparison

As an initial investigation, designs with similar performances (and thus from a single cluster) are analyzed manually. It is questioned to what extent these designs are really different. For this purpose, the design set of cruise ships as designed by *Droste (2016)* is used. Due to the competitive commercial industry of cruises, the main relevant performance characteristics are their cost performances. Therefore in this section a selection is made based on the cost and operating expenditures (OPEX) of the designs, see Figure 13. Six designs are selected with a cost of 57 MEUR and an OPEX of 24.5 MEUR, which are plotted in Figure 14. First the designs are compared based on direct physical attributes, and then based on numerical analysis of their chromosomes.

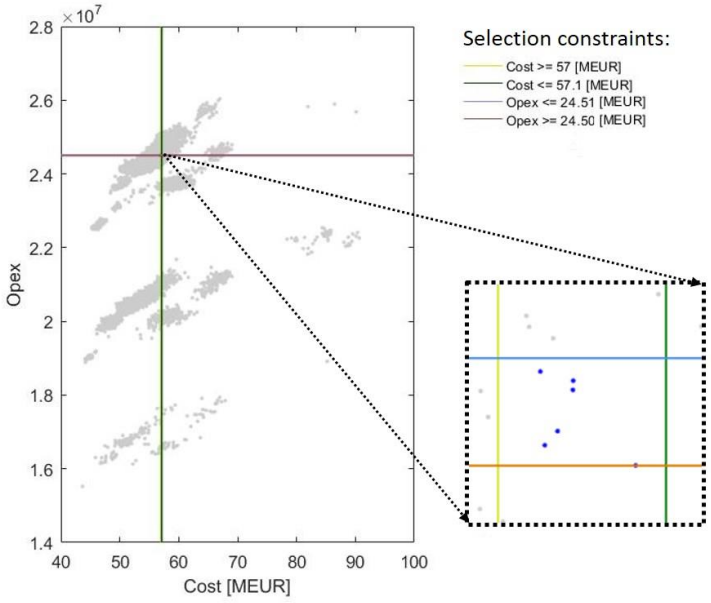


Figure 13: Cost versus OPEX plot of all designs resulting from the packing algorithm as used in *Droste (2016)*, including the selection constraints for the six designs in Figure 14.

2.1.1 Compare physical attributes

Initially the designs are compared based on their main dimensions, which are listed in Table 1. It reveals that regarding these main dimensions, the designs are quite similar. This was to be expected, since cost and displacement are often heavily correlated, and the designs were selected to have the same price. The only notable differences are that design F is a bit longer and narrower, and that the GM values fluctuate a bit. For designs A to E, the fluctuating GM reveals that there are differences in the layout, since the only cause can be a change in weight distribution. For design F the lower GM seems to be caused by its higher L/B.

Table 1: Table of the main properties of the six selected cruise ship designs

Tag	ID	Loa	Boa	T	Displ	Volume	Cost	Packing density	GMt
A	63460	139	19.9	5.92	8828	35974	57.03	0.75	0.5
B	60330	139	20.2	5.86	8844	35695	57.03	0.75	1.0
C	15941	140	20.5	5.79	8878	35277	57.04	0.74	1.6
D	15231	139	20.2	5.85	8854	35700	57.04	0.75	0.8
E	85601	139	20.2	5.89	8902	35721	57.04	0.74	1.1
F	69788	146	19.1	5.90	8862	35741	57.08	0.75	0.2

The layouts of the different designs are plotted in Figure 14. A first thing to notice is that there are no two designs clearly similar. A comparison of the shapes and sizes of the different superstructures is already sufficient to confirm this notion. It is however possible to find similarities based on various properties, especially when assessing the layout:

- The three recreational spaces (restaurant, lounge and theater) are in the same location for designs A and E (encircled with dots).
- Generator room positions, indicated with dark green, are the same for the designs A, B, C and F
- The positions of both hospitals, indicated in red, are near the bridge for all designs.

This list can easily be expanded, and shows that there is no clear way to define the designs as being similar. In fact, while the designs are very much alike regarding their main properties, visually they are all different in various ways.

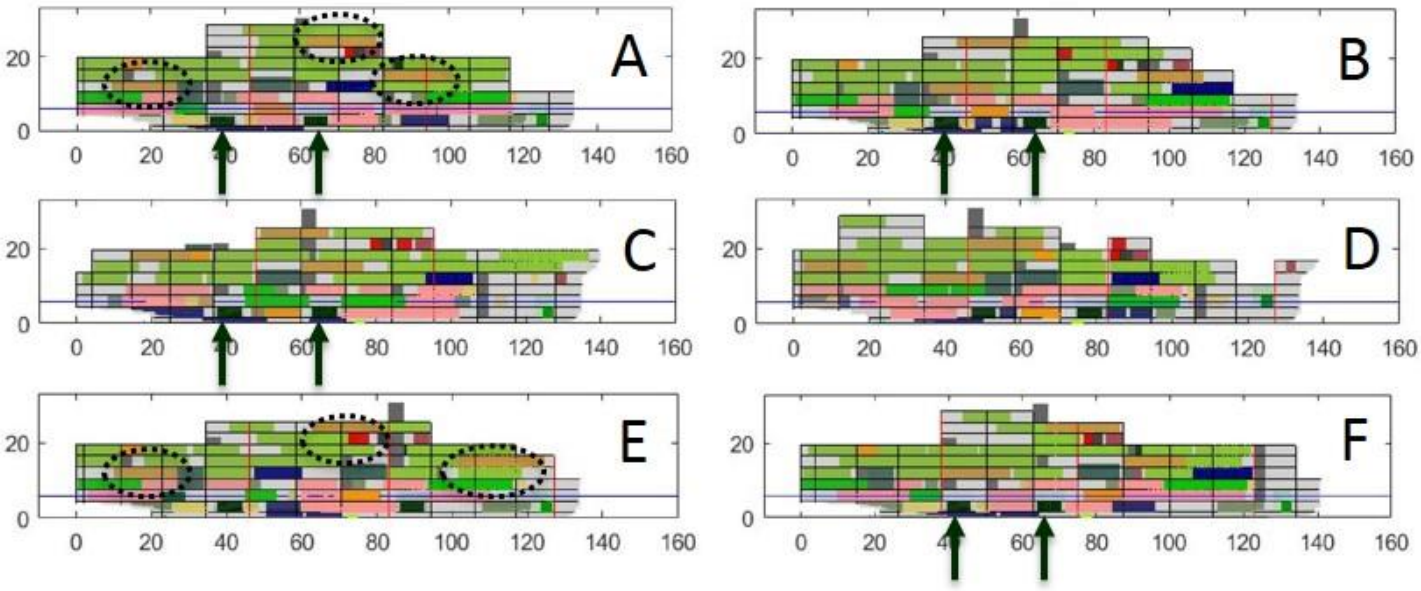


Figure 14: Side view representation of the six selected cruise ship designs

2.1.2 Chromosomal comparison

Since the packing approach uses a genetic algorithm, each design in the packing approach is fully described by a chromosomal representation X, which is a vector of values between 0 and 1. For the cruise ship designs the length of its chromosome N=166. Numerically comparing two of these chromosomes could then be obtained by calculating the distance between the chromosomes in N-dimensional space. A large distance would ten correspond to designs being very different. Therefore as a first step in comparing two chromosomes, a normalized version of the city block distance is calculated:

$$\frac{3}{N}\sum_{i=1}^{N}\left|X_{1,i}-X_{2,i}\right|$$

The factor 3/N is for normalization, since if the two chromosomes would be created completely random, the expected value for the proposed sum would be N/3. A value of about 1 (or higher) would therefore indicate that the chromosomes are as different as randomly generated chromosomes while a value of 0 means that the chromosomes are identical. The results listed in Table 2 show that, while all values are drastically lower than 1, designs can be divided into three groups; A, B, E and F form one group and designs C and D form two individual groups.

Table 2: Comparison of the designs by normalized city block distance

Normalized Σabs(Δxi)	A	B	C	D	E	F
A	0	0.18	0.54	0.58	0.29	0.18
B		0	0.55	0.57	0.24	0.16
C			0	0.47	0.57	0.53
D				0	0.62	0.56
E					0	0.22
F						0

Next is investigated how the designs are related to each other in the family structure of the genetic algorithm, but the problem is that this information is not directly available in the data. This is attacked by looking at the number of genes that are exactly the same, since when a new offspring is generated, they will have approximately half of their genes exactly match the genes of both parents. In Table 3 gives the percentage of fully equal genes between the different designs, which is called the Hamming similarity (inverse of the Hamming distance as defined by *Hamming (1950)*). The result looks a lot like Table 2, showing that the four designs A, B, E and F are probably 1st order related (direct parents children relations), while designs D and C are probably 4 or 5 generations detached from all other designs.

Table 3: Comparison of the designs by Hamming similarity

Percentage equal xi	**A**	**B**	**C**	**D**	**E**	**F**
A	100%	58%	4%	1%	41%	53%
B		100%	4%	3%	45%	56%
C			100%	4%	5%	5%
D				100%	3%	3%
E					100%	49%
F						100%

The similarity metrics used in this section give a much better grasp on the comparison of the designs. It for instance gave a fair argument how the designs can be divided into three different groups (group A-B-E-F, group C and group D). Furthermore although all designs were very close to each other in Figure 13, they are still different.

2.2. Set comparison

At the TU Delft, next to the design set of the cruise ship model from the previous section, a design set from a mine counter measures vessel (MCMV) is available, which was created in *Duchateau (2016)*. An example of one design is displayed in Figure 15 including an explanation of its objects. The goal in this section is twofold. First an initial assessment of the composition of these design sets is made. This could aid in understanding emerging families in the rest of this thesis. A second goal is to select one of these sets which can be used for a test case.

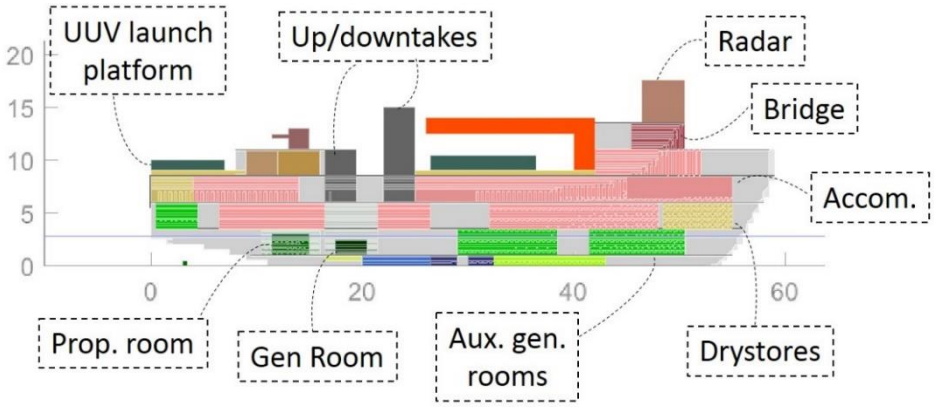


Figure 15: Example of a MCMV design from *Duchateau (2016)*

The number of designs is in the same order of magnitude for both sets. The cruise ship design set has about 22000 designs, while the MCMV design set has just over 17000 designs. Regarding the columns of the datasets, there are many differences in both design characteristics as performance attributes, which makes comparison difficult. However, they do share the property that both designs are fully represented by a chromosome, which is thus a logic starting point for the comparison.

Table 4 shows the composition of the chromosome for the MCMV model. It contains 106 genes, of which $43x2 = 86$ genes correspond to the initial x- and z-positions of its 43 objects[7]. The remaining 20 genes consist of:

- 3 for varying requirement settings
- 6 variables define the envelope
- 3 for varying deck heights
- 4 define the positions of the bulkheads
- The last 4 genes are switches influencing the layout:
    - Whether a gun is present, and its firing direction
    - Whether there are Davits on deck
    - How the uptake is connected to the engine.

The chromosome defining the cruise ship model on the other hand consists of 166 genes. Although this is just about a 50% increase in the amount of genes, the amount of variation in the design space increases exponentially regarding its dimensionality. The design space is therefore less likely to be sufficiently explored. For this model, $75x2 = 150$ genes define the initial positions of the 75 objects. The other 16 genes correspond to the remaining genes discussed for the MCMV, but without any genes defining switches for the layout.

[7] When the objects positions are initialized, the packing algorithm starts shifting these objects to positions so that they fit inside the hull, and do not overlap with other objects.

Table 4: Composition of the chromosome as defined for the MCMV model

Group	Number	Component	Remark
Req.	1	V_max	
	2	V_cruise	
	3	Range	
Envelope	4	Hull type	always the same
	5	L_oa	
	6	T	
	7	B	
	8	Bow factor	
	9	Stern factor	
Decks	10	Tank top height	1 or 1.5m
	11	Deck height	always 2.5m
	12	Damage control deck	deck 4 or 5
	13-16	**2 objects x,z**	
Bulkheads	17	Number bulkheads	Cosine bulkhead placing
	18	Minimal distance	
	19	fwd_l_fact	
	20	aft_l_fact	
	21-28	**4 objects x,z**	
	29	Uptake side switch	
	30-43	**7 objects x,z**	
	44	MRSV Davits on/off	
	45-48	**2 objects x,z**	
Gun	49	Gun on/off	
	50	Firing direction	
	51-106	**28 objects x,z**	

The next step in comparing the design sets is examining histograms of the gene values. In Figure 16 and Figure 17 six examples are displayed of respectively the cruise ship and the MCMV. In all histograms there are one or more peaks present, which is due to using a genetic algorithm in the packing approach. The genes from Figure 17 a-c and f, have a relatively high variety with multiple peaks, which do not exceed 20% presence. The genes from the cruise ship on the other hand show a much smaller variety, with peaks often exceeding 40% presence. It seems therefore that for the cruise ship design the packing algorithm remained searching in the same region, opposed to the MCMV data.

However, the genes from Figure 17 d-e also show high peaks. Investigating all 106 genes form the MCMV showed 7 genes with such peaking behavior. It appeared that these peaks were caused by a variety of reasons:

- After synthesizing a design from a chromosome, the genes corresponding to z-positions of objects are fixed to correspond the final position of the object. This correction is implemented in order to save run time of the packing approach[8]. But

[8] The bin packing algorithm will have to check less available positions for the offspring of a design if the initial

since the up- and downtake have a limited number of possible z-positions, they show peaks.

- Three genes are overwritten in the code, and are therefore never used.
- Two genes correspond to the x- and z-positions of the main gun. But when there is no gun present, the values for the position is set to zero, which causes peaks.

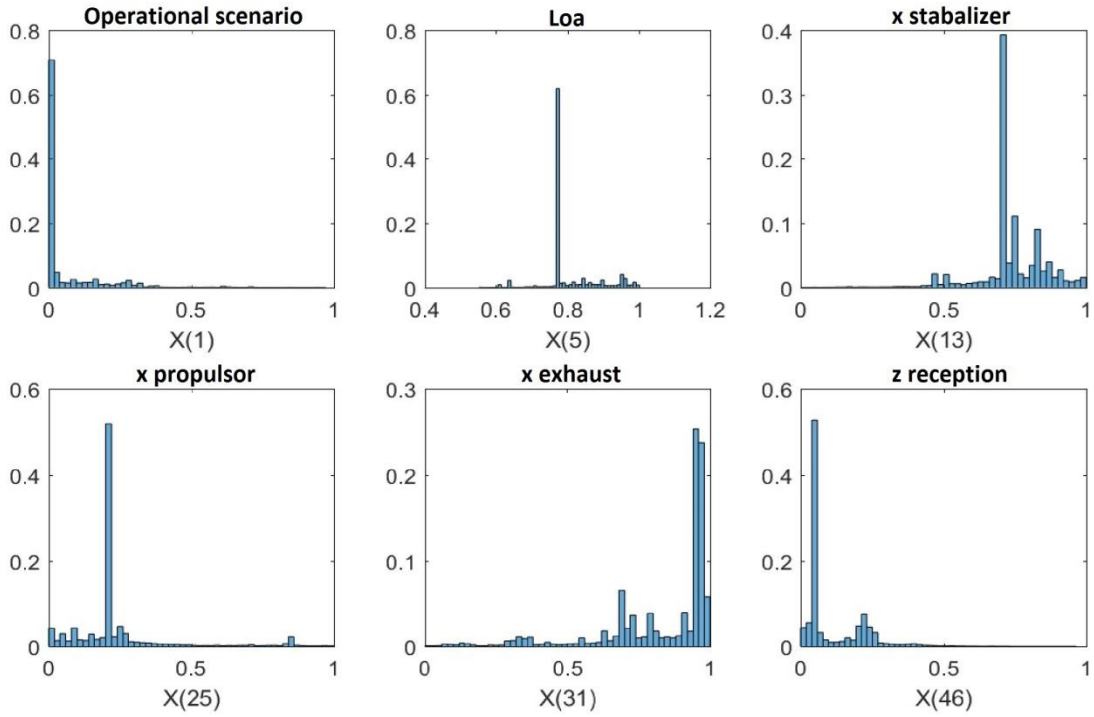


Figure 16: Six histograms of the gene values for the cruise ship

positions of the objects are closer to their final position.

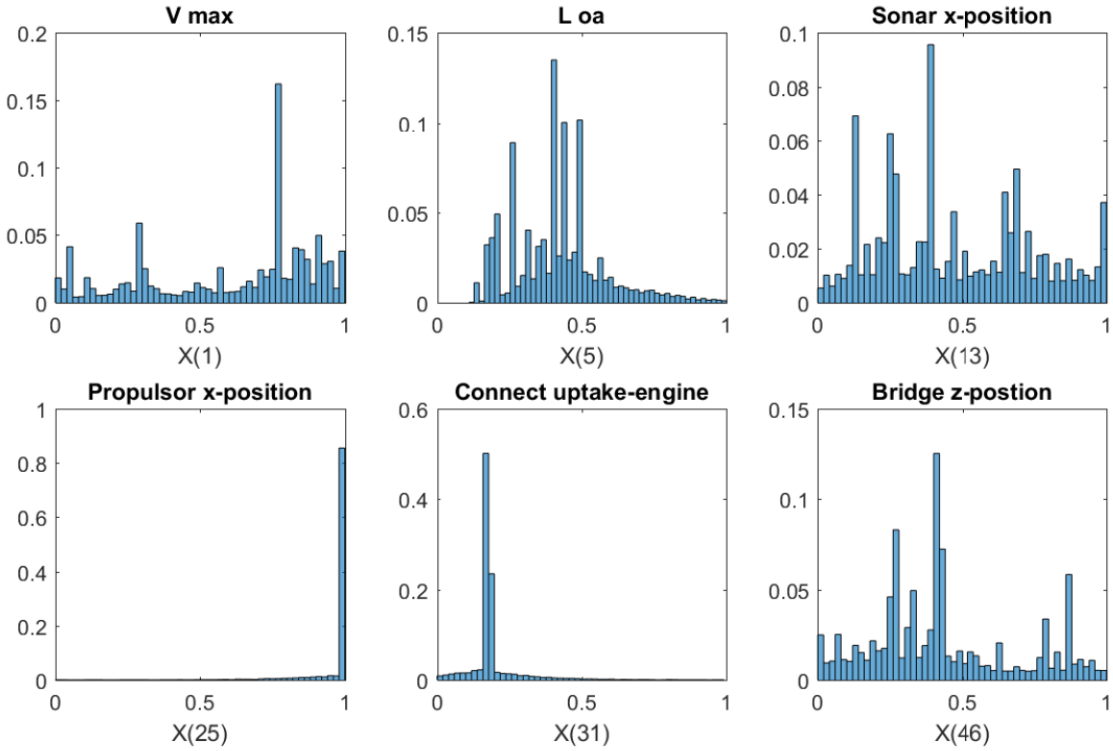


Figure 17: Six histograms of the gene values from the MCMV

Wrapping up, the MCMV dataset seems to have a higher diversity in the design space due to both a lower number of genes and gene values with smaller peaks. Therefore this design set will be used for the test case presented in chapter 4. The analysis of gene values furthermore showed, that there are a number of discrete input variables present which could cause families in the data. These are:

- Tank top height is either 1 or 1.5m
- Damage control deck is either deck 4 or deck 5
- The presence of the main gun, and if it is on the bow or the aft (corresponding to its firing direction). This is illustrated in Figure 18.
- The presence of Davits.

Section 2.1 showed that it is hard, when looking at different designs, to assess which designs are more similar than others. The chromosomal comparison did show however that looking at their numeric values, and calculating distance metrics, handed a more structured approach to make a comparison. Additionally these distance metrics can be calculated fast, and could therefore be helpful to quickly compare the tens of thousands of ship designs within one design set. How this concept is used to find families, is further elaborated in the next chapter.

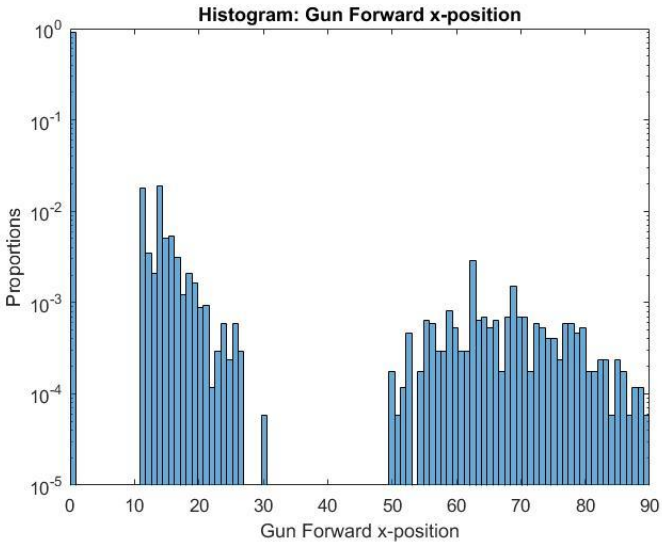


Figure 18: Logarithmic histogram of the x-position of the main gun. Three families emerge: no gun, gun on the aft or at the bow.

3. Method

In order to select a decent approach for dividing the set of designs resulting from the packing approach into families, various fields of science corresponding to similar problems were listed and investigated, such as:

- Big data
- Pattern recognition
- Machine learning
- Markov theory
- Network theory
- Neural network theory
- Bayesian statistics

It finally turned out that all these fields of science share a common ground in being related to "clustering algorithms". The first three fields use clustering algorithms for various purposes, whereas the last four fields serve as mathematical foundations for a variety of clustering algorithms. Examples are the application of Markov random walks *Meilă and Shi (2000)* or SOM analysis based on neural networks *Kohononen (1990)*. Therefore the method of choice in this thesis is found to be clustering algorithms themselves. These algorithms, as their name suggests, are devoted to find clustering structures in data. An example of their application is in companies as Facebook and Google where people are divided into clusters to achieve better assessment of which advertisement suits which person best *Schutt and O'Neil (2013)*. The analogy is that in this case the designs are divided into clusters to achieve better assessment of which design decisions suits which performance requirement best. Assuming that the function mapping the design space to the performance does contain a trend (i.e. is not completely random), the more distinct the clusters are, the more probable they reveal knowledge of the relation between design and performance space.

A problem is however, that there is no clear notion of what 'being distinct' means. Studying for example the designs in Figure 19, there are various ways of comparing them. Looking at main dimensions designs A, B and C are similar, while design D is a bit longer. Whereas looking at the position of the working deck designs A and C have a working deck amidships, while at designs B and D it's positioned at the stern. Finally if assessing the main gun, only design A has one, while it is absent in the rest of the designs. These examples show that clustering is inherently a subjective science, as there is no single right or wrong way to cluster any given data *Theodoridis and Koutroumbas (2009)*. It is therefore important to investigate various sensible ways of clustering the set of designs, in both design features as applied clustering methods.

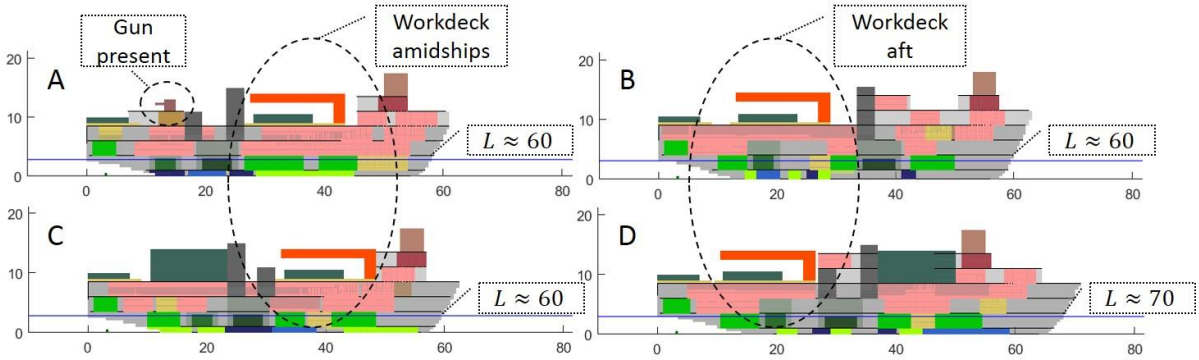


Figure 19: Four MCMV-designs resulting from the packing approach, where various differences and similarities are pointed out.

Clustering is a six step process, see Figure 20, which is described in *Theodoridis and Koutroumbas (2009)*. In Figure 20 the six step process is illustrated using LEGO blocks. The reason is that it is very common in clustering to set up the problem in various different ways, in order to extract multiple results. The LEGO blocks thus represent the iterative nature, where multiple block stacks can be made with its own results and conclusions. The steps including their specific algorithms used in this method are discussed more in-depth below.

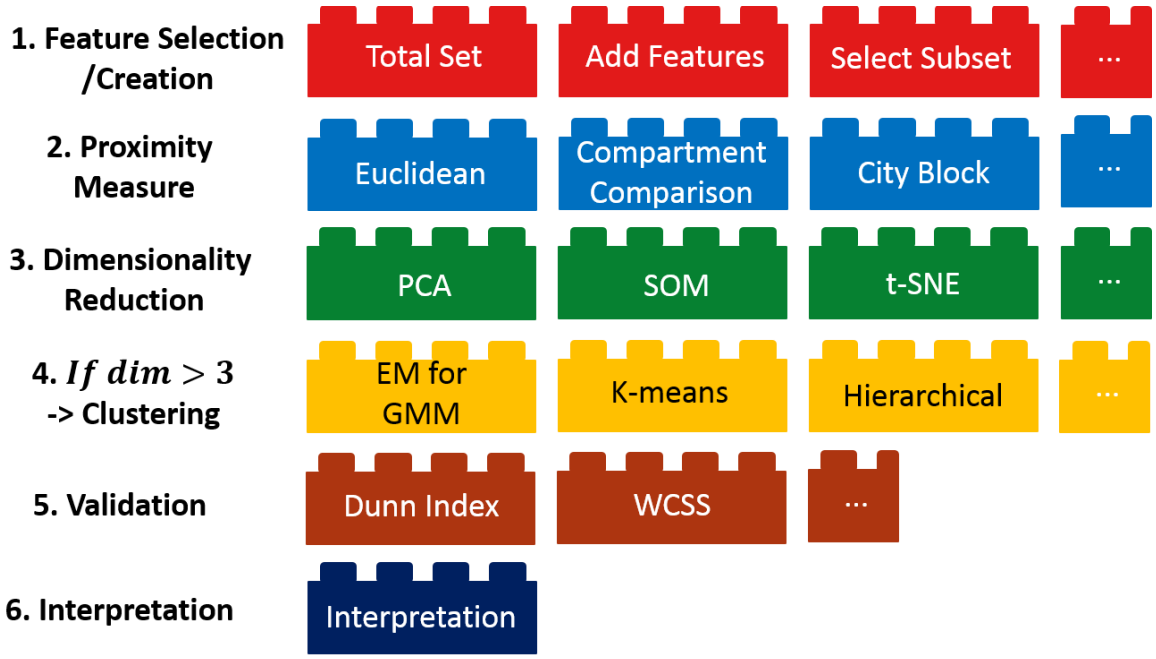


Figure 20: Six step clustering method visualized with LEGO blocks. Every block stack is a new method on its own.

3.1 Feature selection/creation

As illustrated in Figure 19 which features are considered important, greatly influence the families that will emerge. Therefore the features which are important for the specific analysis should be selected. This could concern the total set of features, or maybe only a subset of the features is of interest. Examples of potential interesting subsets are features concerning the layout of a design, all performance features or features regarding the global parameters of the

designs. It could furthermore be necessary to add new features. An example of a new feature is adding displacement, when L, B, T and c_B are available.

Next it is very important in this step to consider whether the selected features should be normalized/standardized. In general, when the features use different units and scales, they should be standardized. The reason is that for most distance metrics, which will be defined in the next section, a comparison is made between the absolute values of the variables. For instance if one variable is displacement it has typical values in the order of thousands [ton], while the draft has typically values in the order of maximum 10 [m]. This would thus mean that variations in displacement would always dominate variations in draft, which is unwanted. Standardization of a variable can be obtained by taking the z-score. This sets its mean to zero and its standard deviation to one, with the following transformation:

$$z_i = \frac{x_i - \mu_x}{\sigma_x}$$

For all data points x_i.

3.2 Proximity measure

The proximity measure defines how similarity between data points is measured by selecting a distance metric. In selecting a distance metric, it is first important to check which distance metrics are able to coop with the type of features. For instance if the features are solely binary, the Hamming distance is probably a good option, since it measures the number of equal symbols. An example would be if the designs have a number systems which are optional, where the variables for the presence of these systems are 0 when the system is absent, and 1 when the system is present. Hamming distance would then measure the number of corresponding systems on board of two designs, which seems a valid comparison. Hamming distance is thus normally a bad choice for real valued features, except if there is a reason for the symbols to be equal. Which is true for vectors resulting from a genetic algorithm, as is used in the packing approach as shown Table 3 in chapter 2.1.

The features resulting from the packing approach are in general real valued. The most common distance metrics are Minkowski distances like Euclidean (p=2) and city block (p=1) distance:

$$d(x, y) = \sqrt[p]{\sum |x_i - y_i|^p}$$

Euclidean distance measures the shortest path (straight line) between two points, whereas city block corresponds better to walking distances. Other interesting metrics concerning real valued features are cosine similarity and Mahalanobis distance *Theodoridis and Koutroumbas (2009)*. Cosine similarity measures equality between the proportions of the scalar components of the data points, and Mahalanobis distance is a special normalized form of Euclidean distance. In the rest of this thesis Euclidean distance is used.

3.3. Dimensionality reduction

Reducing the dimensionality of the data means that the same data is represented with less features. An example is a dataset with variables L, B, T, c_b, and displacement. This 5-dimensional data could also be represented by 4 variables, since displacement equals the product of the others.

There are various techniques that automatically find these structures, and reduce the dimensionality of the data. Examples are principal component analysis (PCA), self-organizing maps (SOM) *Kohononen (1990)* and t-distributed stochastic neighbor embedding (t-SNE) *Maaten (2008).* PCA creates new features that are linearly depending on the old features, whereas SOM and t-SNE project the data in a non-linear way. Dimensionality reduction serves two main purposes:

1. It improves the quality of the clustering algorithm as is shown for PCA by *Ding and He (2004).*
2. When the amount of features is reduced to two or three, the result can be used as an initial visualization of the problem.

In this thesis PCA is used, and is therefore further elaborated below.

3.3.1 Principal Component Analysis

PCA is a valuable technique for exploratory analysis of high dimensional data. It rotates the original dataset in such a way that the first principal component (pc) corresponds to the direction with the highest variance, the second pc is orthogonal to the first pc and contains the second highest variance, and so on. A two-dimensional example is shown in Figure 21. This is useful for a number of reasons. Most important is that the amount of variance can be interpreted as being the amount of information *Linsker (1989).* This reveals how PCA can be used for dimensionality reduction: Selecting and examining only those first couple of pc's that have the highest variance.

Since it is only possible to plot up to three dimensional data[9], a plot of the first three pc's will show you as much information as possible in one plot. On the other hand interpreting the content of the plot gets harder due to the complex values on its axis, since each pc is a linear combination of all input features (for instance, pc_1 could be equal to 0.5L+0.2B-0.85DWT) . But the focus in this thesis lies in identifying the multidimensional structure (clusters) in the data, which will still be visible in the plots. In fact, if there is a direction in space where clusters do show up, this direction has an increased probability of having a high variance, which makes it more likely to end up in the first three pc's *Ding and He (2004).* This property is illustrated in Figure 22.

[9] Higher dimensional plotting is technically possible (i.e. using colour and/or time), but the same argument holds.

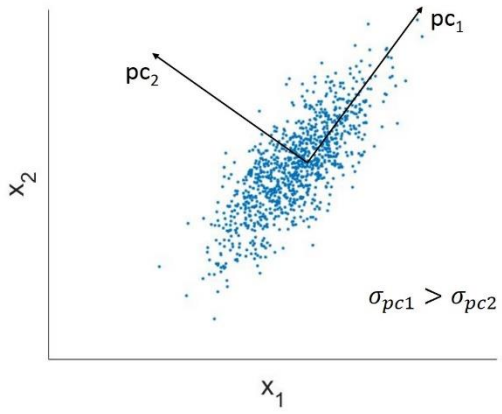


Figure 21: Illustration in 2D how PCA rotates the data. It makes it as "flat" as possible.

Other valuable information resulting from PCA is examining the directions of the first couple of pc's. This reveals both the importance of the various features and how the features are correlated. An effective way of visualizing this information is with a biplot, where the amplitude of every feature is plotted for the first two pc's, which will be used in this thesis.

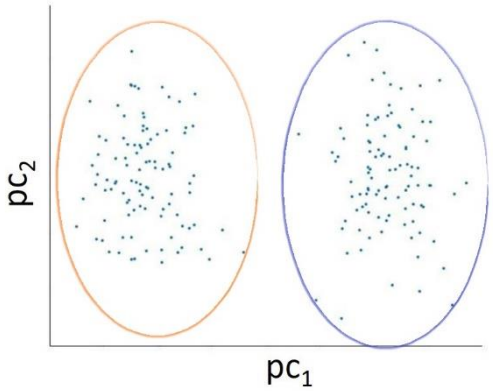


Figure 22: PCA applied to the data used in Figure 5. The first pc lies in the direction that reveals the clusters.

3.4. Clustering

There are numerous clustering algorithms available, which can roughly be divided into two main groups *Jain (2010)*: partitional and hierarchical methods. This division is illustrated in Figure 23. Hierarchical algorithms return a hierarchical tree structure (called a dendrogram) where every branch divides the set into smaller clusters. Partitional algorithms on the other hand return a partition of the dataset where every point is (partly) assigned to a certain cluster. Partitional algorithms can in turn be divided into hard and fuzzy methods, where fuzzy methods can assign points partly to multiple clusters at the same time. Since the goal in this

thesis is to find clear distinct families of ship designs, hard partitional methods are emphasized. Particularly the k-means algorithm is used, which is discussed in the next section.

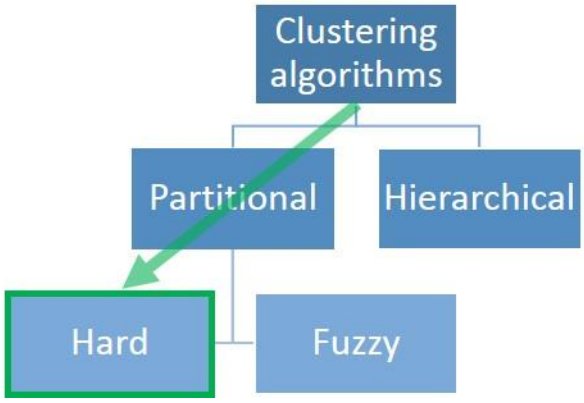


Figure 23: Hierarchical chart showing a coarse division of the different type of clustering algorithms.

Note that in this thesis the goal of applying clustering algorithms is to identify structure in the data that is not identified yet. Therefore only if (after dimensionality reduction) the number of features exceeds three, applying a clustering algorithm is useful, otherwise the data can be plotted in order to visualize the structure.

3.4.1 K-means

The k-means algorithm, which exists over 50 years, is still one of the most popular clustering algorithm used these days. It is therefore selected in most machine learning courses as one of the basic algorithms *Schutt and O'Neil (2013), Fox and Guestrin (2015), Leek et all. (2016).* Main reasons for its popularity are its ease of implementation, simplicity, efficiency, and empirical success *Jain (2010).* Next to these arguments, the reason for picking k-means over the other well-established hard partitional clustering algorithm DBSCAN is twofold:

1. DBSCAN discards noise, which is not present in the dataset at hand. Every design should be assigned to a certain cluster, which is the case with k-means.
2. K-means is convenient for its standard implementation in MATLAB.

The downside of k-means is that it requires the amount of clusters (k) as input. Therefore in practice the algorithm is applied to the dataset for various values of k, where the validation step reveals the most promising value. The algorithm works as follows:

- Initialize by selecting k distinct center points $\boldsymbol{c_1}, \cdots, \boldsymbol{c_k}$ in space.[10]
- Repeat until convergence:
  - Assign every data point to cluster i if it is closest to center point $\boldsymbol{c_i}$
  - Shift every $\boldsymbol{c_i}$ to the center of mass of the data belonging to cluster i

[10] Various initialization methods exist such as k-means++, which is used in this thesis. The easiest is randomly selecting distinct positions as is used in the example of figure 7.

This process is illustrated in Figure 24. In Figure 24a the data itself and the random initialization of the centers is displayed. The first and second iterations are shown in respectively Figure 24b-c, and finally convergence is reached in Figure 24d.

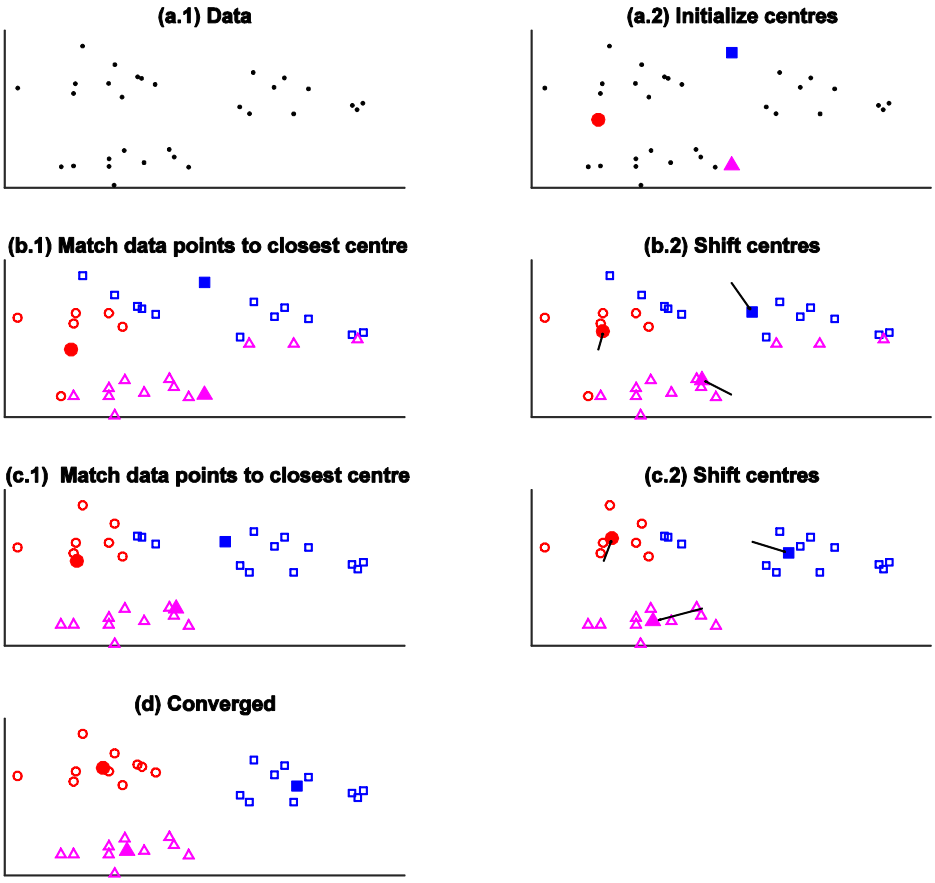


Figure 24: Illustration of how k-means converges on a 2D artificial dataset for k=3

The result of k-means is depending on the way it is initialized. Therefore it is common to let the algorithm run multiple times and pick the best result, where the best result is defined as the result with the lowest WCSS (see section 3.5.1). In this thesis all results are obtained by running k-means 50 times, which was empirically verified to give stable results. Furthermore there are various initialization methods. For example in Figure 24 the algorithm is initialized by simply placing the centers at random points in space. In the rest of the thesis, k-means++ is used as initialization method. K-means++ iteratively allocates positions of data points as initial centers, but with increased probability of acceptance if it is far from the other allocated centers. It improves the quality of the results while the runtime remains similar *Arthur and Vassilvitskii (2007).*

3.5. Validation

Validating the clusters serves two purposes. First, it is not yet clear whether there is a clustering structure present in the data. Validating if this is the case is not trivial since there are more than three dimensions, which makes it hard to visualize. Second, since k-means is applied for various values of k, the number of clusters is still to be determined.

Validation is performed by analyzing appropriate metrics that indicate the quality of clusters. The two metrics used in this thesis are discussed below.

3.5.1 within cluster sum of squares (WCSS)

The first metric used for validation is the within cluster sum of squares (WCSS), which, as the name suggests, sums the squared distance of every data point to its corresponding cluster center *Jain (2010)*:

$$WCSS = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \boldsymbol{c_i}\|^2$$

Where $\boldsymbol{x}$ denotes a data point, and C_i denotes the set of data points belonging to cluster i.

The reason why this metric is used, is because it is a direct result from k-means. In fact, k-means is a heuristic optimization algorithm that minimizes the WCSS. For increasing k, the WCSS is in general decreasing with WCSS = 0 for k = #data points, since every data point then has its own center. Thus no conclusions can be drawn from a single WCSS value. On the other hand, looking at the curve when plotting the WCSS versus k does reveal information. When a knee appears, this is an indication that the data contains the number of cluster corresponding to the position of the knee, see Figure 25.

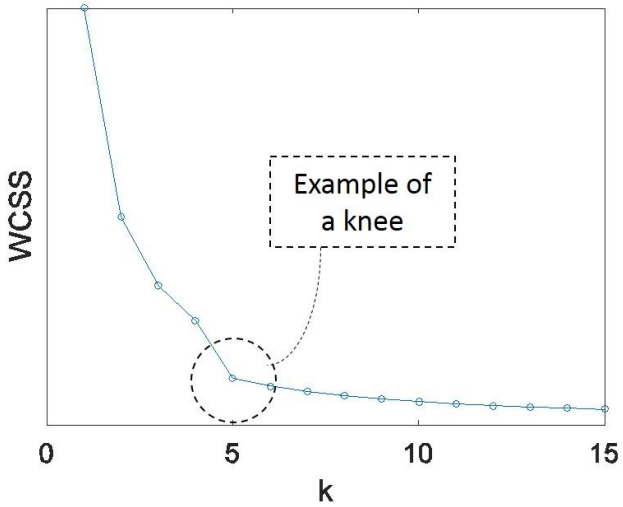


Figure 25: Example of a knee in plotting the WCSS versus k at k=5. This indicates that the data contains 5 valid clusters.

3.5.2 Dunn-index

This thesis seeks clusters based on physical attributes, which means that certain combinations of features result in infeasible designs. Thus, opposed to density clusters, there should be real gaps in-between the clusters. Therefore the Dunn-index is an appropriate metric to validate the clusters, since it measures the size of the gap *Theodoridis and Koutroumbas (2009)*. It is defined as:

$$Dunn\ index = \min_{\forall i, \forall j \neq i} \left(\frac{d(C_i, C_j)}{\max_{\forall k}(diam(C_k))} \right)$$

Where $d(C_i, C_j)$ is the minimum distance between points from clusters C_i and C_j, and $diam(C_k)$ is the maximum distance between points from cluster C_k.

To be more precise the Dunn-index is a measure that gives a lower bound for the distance between the clusters relative to the size of the clusters. A Dunn index of 1 or higher would therefore mean that the minimum distance between the clusters is higher than the diameter of the biggest cluster. The Dunn index does not exhibit any trend with respect to k, and therefore the highest value would indicate the number of clusters present.

3.6 Interpretation

When the result is deemed to be valid, the most interesting part is to interpret the result. This is where the information (i.e. the structuring of the data in clusters) is interpreted to generate knowledge and wisdom. There is no roadmap telling how to do this, but an initial step is trying to correlate the clusters to discrete features in the dataset. In order to test whether the method in this section is able to elucidate families of designs from the data of the packing approach, it is applied to a test case in the next section.

4. Test Case: Mine-countermeasures vessel

In this chapter it is tested whether the method is capable of finding families of ship designs that result in new knowledge of the relation between the design and performance space. Therefore the method is applied to the dataset of the MCMV as developed by *Duchateau (2016)*.

4.1. Survivability of machine systems

In naval ship design, an increasing value is granted to the assessment of survivability in early stage ship design. In that mind, a current PhD position at the TU Delft is devoted to incorporate this assessment on the designs resulting from packing. A first emphasize lies specifically in investigating the survivability of machine systems. Since there is no automated metric for survivability yet, it is initially attempted, to find families of designs in the MCMV dataset regarding its machine systems. If these families are present, this could serve as a data reduction method, allowing an initial assessment of the survivability per family.

In order to asses which systems are of main importance for assessing the survivability of machine systems, their logical architecture[11] of the machine systems is composed in Figure 26. The way in which the systems are related:

- The generator room has to supply electrical energy to the propulsion room, radar and main gun
- Information from the radar is needed to properly fire the main gun
- The propulsion room is connected to the propulsor via the shaft

This thus motivates that the positions of these five objects are used as the features of interest in this section. Since the designs are generated by 2.5D packing, y-positions do not vary for these objects, and thus only the x- and z-positions are selected *van Oers and Hopman (2012)*. This results in a total of 10 features, which are standardized by taking their z-scores.

[11] Terminology as used by *Brefort et all (2017)*

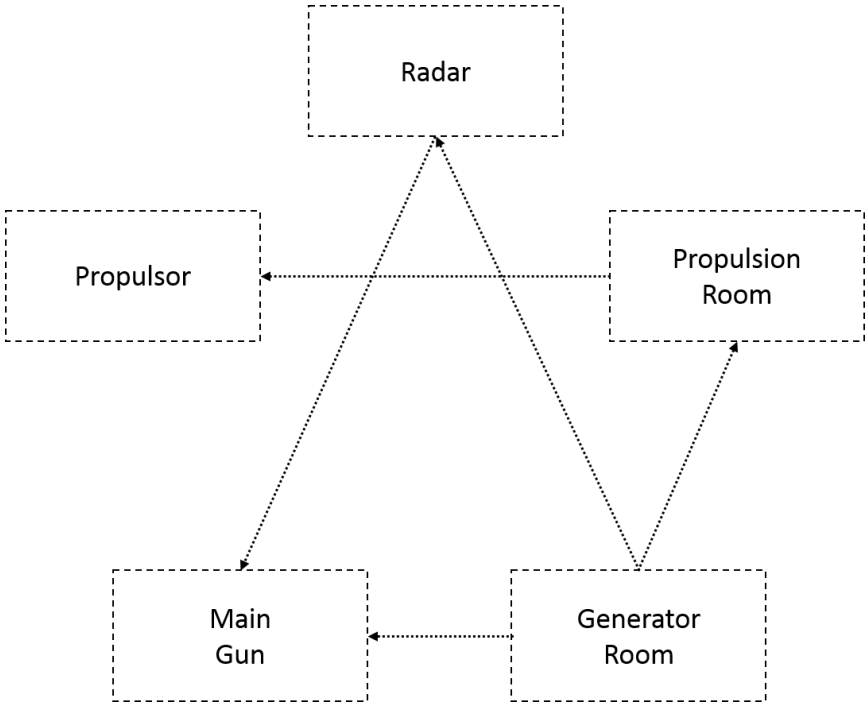


Figure 26: MCMV objects related to the survivability of machine systems and their interrelations

4.1.1. Iteration 1

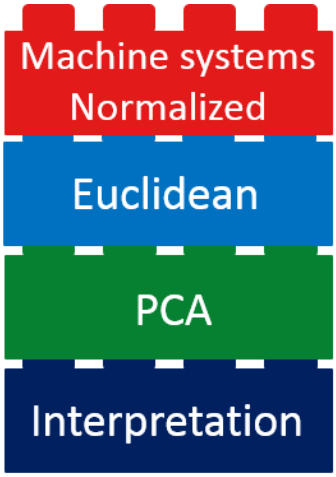


Figure 27: Clustering method used in this section

An initial application of the clustering method to the normalized machine systems dataset is shown in Figure 26. The clustering and validation steps are in this first iteration omitted since PCA is used to reduce the dimensionality of the data to only two dimensions. Figure 28 shows that the first two pc's, upon which the data will be projected, contain almost half of the total variance. How these two pc's are constructed is displayed in the biplot of Figure 29. For instance pc1 is constructed as:

$$pc_1 = -.32x_{prop.room} - .37x_{gen.room} + .00x_{propulsor} + .50x_{gun} - .37x_{radar} + .12z_{prop.room} - .10z_{gen.room} + .00z_{propulsor} + .58z_{gun} - .03z_{radar}$$

A variety of information can be extracted from the biplot:

- Both x and z components of the propulsor are not apparent in both pc's. This is due to the fact that the position of the propulsor is fixed, which also explains why in Figure 28 the first 8 pc's explain 100% of the total variance.
- Both x and z positions of the gun are heavily positively correlated and greatly influence the first pc. The correlation is caused due to the fact that these values are both set equal to zero when there is no gun present on the vessel.[12]
- The second pc mainly consists of z-positions, which explains the result of Figure 31.

[12] When only designs with a gun are considered, these two variables have a negative correlation coefficient equal to -0.48.

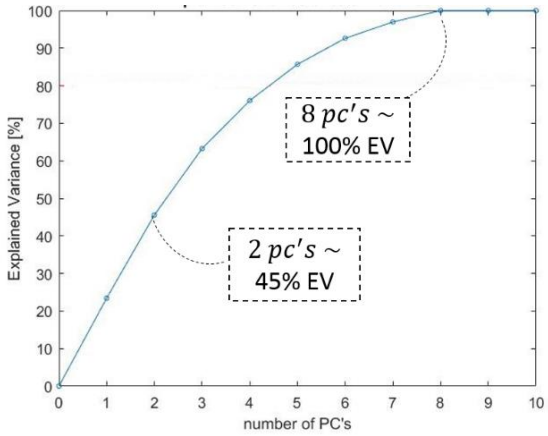


Figure 28: PCA result: Explained variance vs. the number of pc's

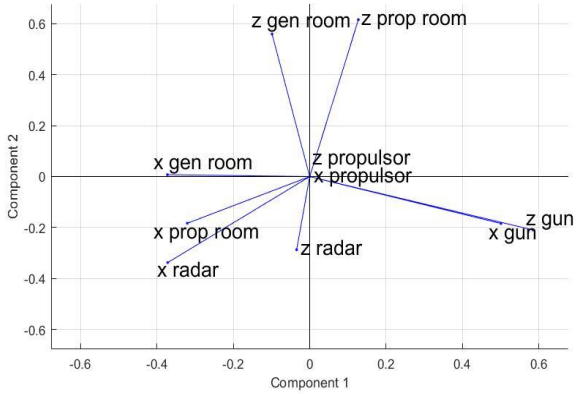


Figure 29: Biplot showing the factors of the first two pc's

In Figure 30 and Figure 31, the data is projected on the first two pc's. Colors are added regarding two discrete input variables which are respectively gun position and tank top height. The structure in the data seems to be dominated by these two input variables. The fact that the gun can be either on the bow or on the stern influences the survivability of machine systems, and are thus two families worth investigating. This result is not new however, since it was coded in upfront (see Table 4). Furthermore the results do not reveal much information about the survivability of machine systems, since:

- Gun x- and z-positions are set equal to zero when there is no gun present, which is basically a definition issue regarding the data. This motivates in next iterations to only select data with either a gun or no gun present.
- Tank top height shifts the global z-position of a number of systems, but the main interest is in their relative position. This motivates to define new z-features in next iterations where the tank top height is subtracted.

First it is tested whether k-means finds other structure in the next iteration. After that, these two lessons learned are implemented in the third iteration.

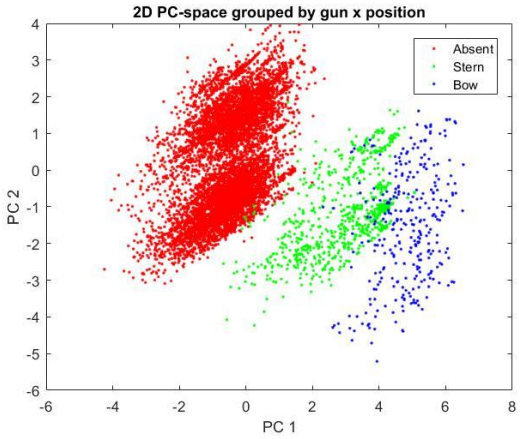


Figure 30: Machine systems data projected on the first two pc's, and colored regarding gun position

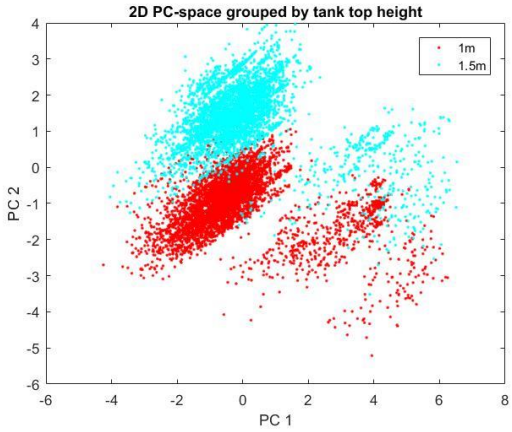
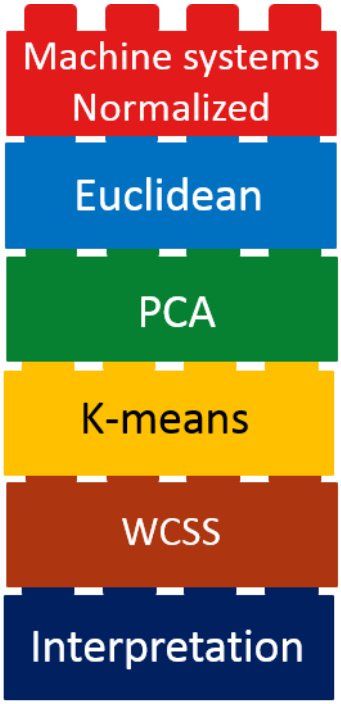


Figure 31: Machine systems data projected on the first two pc's, and colored regarding tank top height.

4.1.2. Iteration 2



The 8 pc's explaining 100% of the total variance are fed into the k-means algorithm. For validation the resulting WCSS values are plotted in Figure 32, which shows a knee at k=3 (thus indicating that the result for k=3 is valid). Looking at the result for k=3 projected on the first two pc's in Figure 33, it shows that the green cluster corresponds to designs with a gun, and the blue and red clusters correspond to designs without a gun with respectively a tank top height of 1m and 1.5m. This indicates that the visibly identified structure from the first iteration corresponds to the structure k-means finds.

Furthermore this observation shows one of the downsides of k-means: it focusses mainly on dense areas in the data. In Figure 33 for k=4 the k-means algorithm divides the group of designs with no gun and a tank top height of 1m into two clusters, instead of finding the clusters regarding the gun position of the bow or stern, visible at the plot for k=5. The reason could be, that since there are a lot of designs in this cluster, the WCSS is reduced more by splitting this denser group. It could however also be the case that there is other structure present, which is investigated in the next iteration.

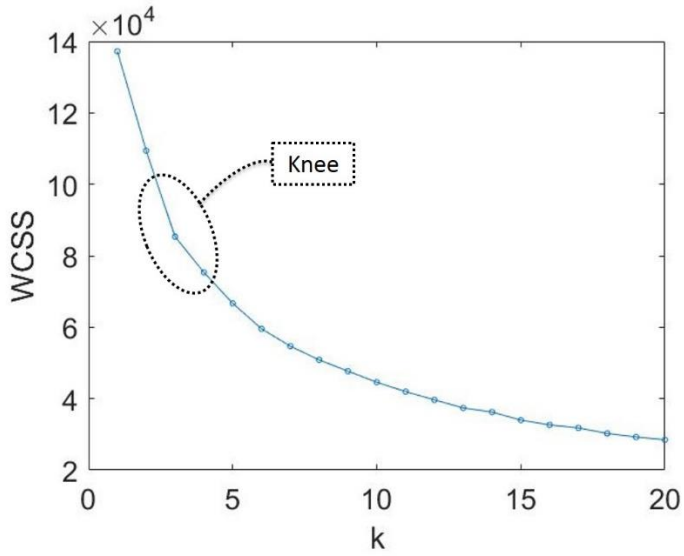


Figure 32: The WCSS for various values of k resulting from k-means, showing a knee at k=3.

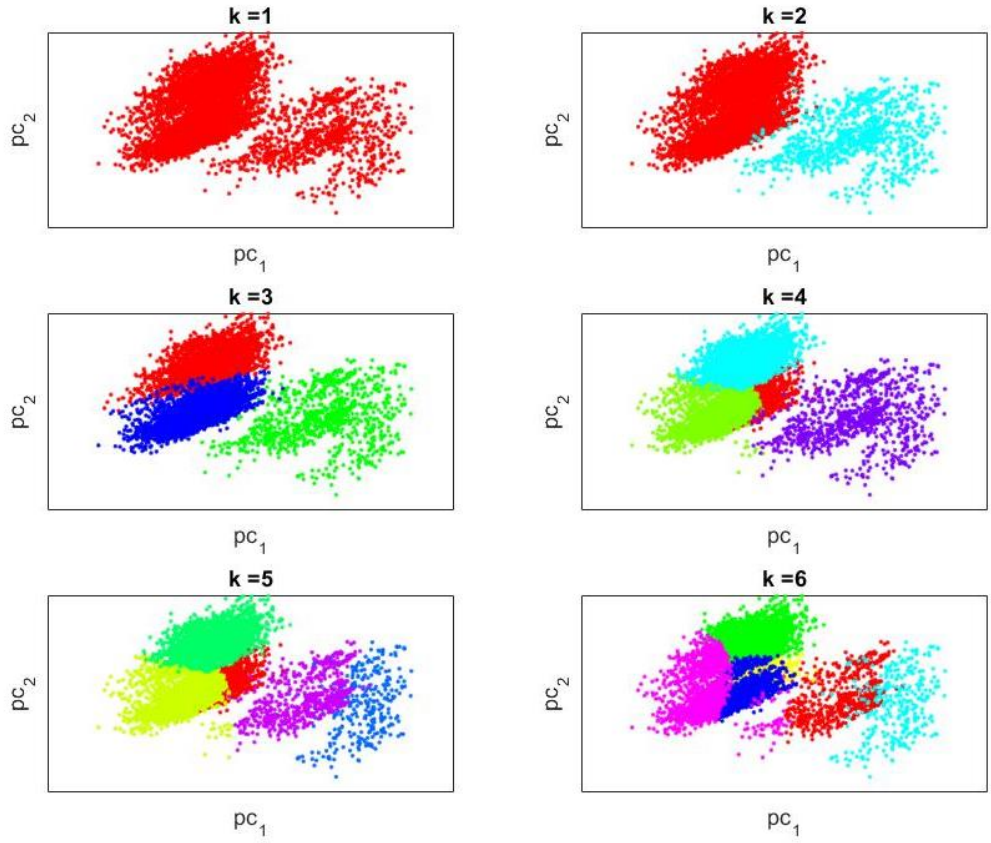
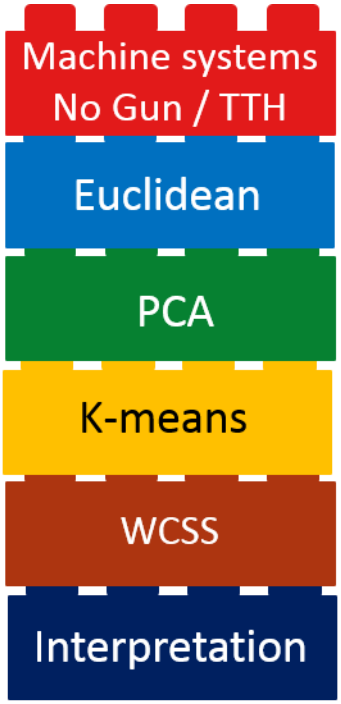


Figure 33: Projection of the data on the first two pc's and colored by the groups resulting from k-means for k=1 to k=6

4.1.3. Iteration 3



As is concluded in the previous iterations, it could be interesting to assess whether there is more structure in the set of designs without a gun. Furthermore the influence of the discrete tank top height is omitted by subtracting it from the z-components of the objects.

Since the position of the propulsor is constant, and there is no gun present on these vessels, 6 features remain to be examined (x- and z-positions of the 3 objects). These features are analyzed using PCA, and the data is plotted regarding its first 2 pc's in Figure 34. Since this figure shows no structure yet, k-means is applied to the data. In order to validate the clusters from k-means, the WCSS is plotted for various values of k in Figure 35. Since there is no knee present in this graph, there is insufficient reason to believe that higher dimensional structure is present.

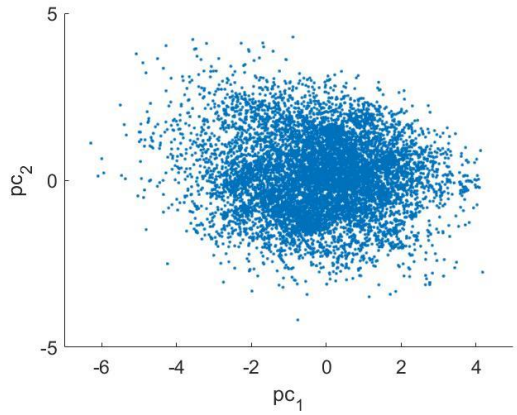


Figure 34: Projection of the data on its first two pc's.

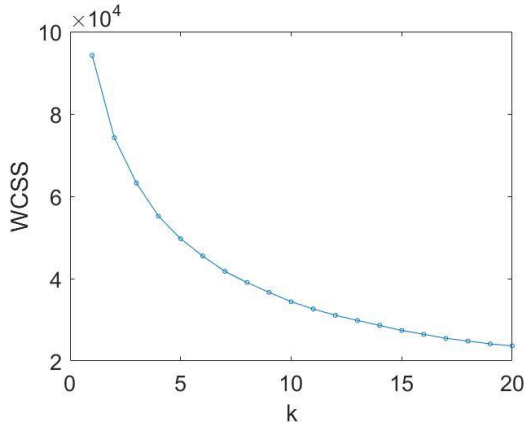


Figure 35: The WCSS for various values of k, resulting from k-means.

The goal of this section was to search for families of designs regarding the survivability of machine systems in order to reduce the data for a naval architect. In all iterations, no other structure appeared regarding the positions of machine systems, than the 3 families of designs that either had no gun, or had a gun with either the gun on the bow, or on the stern of the designs. Since this is a too coarse division to draw any unambiguous conclusions on the survivability of the designs within such a family, this division is not sufficient to serve as a data reduction method on itself.

4.2. Families based on layout

Since naval ships are complex designs that have many interactions between their compartments *Andrews (2011)*, it's particularly interesting to assess families of the MCMV regarding its layout. It is namely expected that these interactions cause many layouts of bad quality, leaving only a finite number of sensible arrangements. An initial indication for the presence of families is illustrated in Figure 36. It shows a one dimensional envelope of 100m, where a generator set and an accommodation block should be placed. Assuming that these must be at least 15m separated due to noise restrictions, two families emerge when plotting x_A versus x_G, corresponding to the permutation of these two objects. Furthermore, the histograms on both axis show that the families can't be identified when looking at them individually. This motivates searching for multidimensional family structures, when looking at a dataset subjective to ten such rules (instead of one), 2 dimensions (instead of one), and 43 different objects (instead of 2).

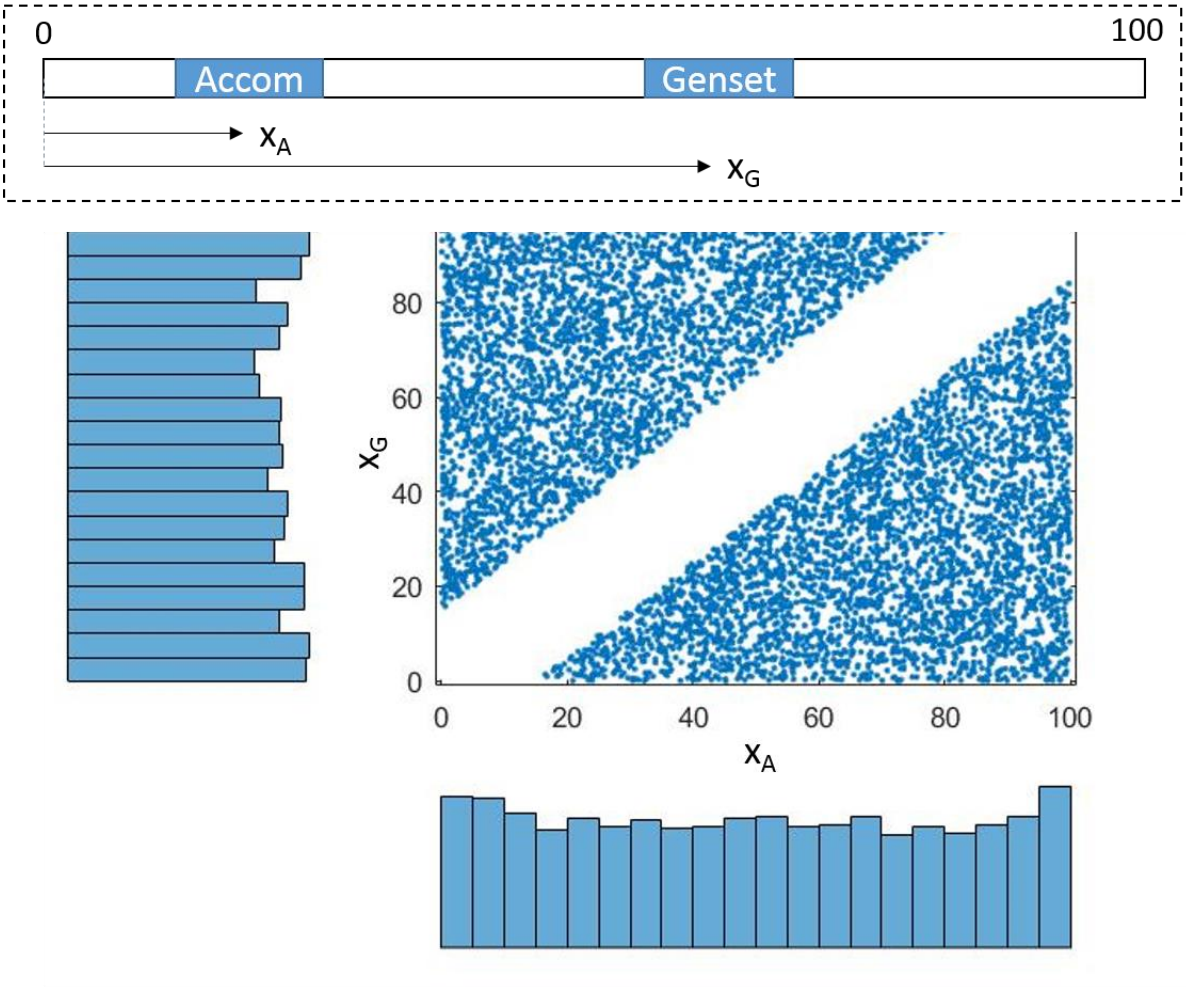


Figure 36: Illustration of existence of families due to connections between the compartments.

The problem is however, that since the packing approach mainly operates as a sizing tool, these interactions are omitted. This means that the compartments are more or less randomly stacked into the hull. Therefore, this section starts with selecting designs based on the quality

of their layout. Then the proposed method is applied for finding multidimensional structures in the layouts of these selected designs. And finally, the results will be discussed.

4.2.1. Design selection based on designer rationale

The goal in this section is to select the top 10% of designs with the best layouts based on designer rationale. The layouts of the resulting subset of designs are then assumed to contain more structure than the remaining 90%. This enables verification of the hypothesis that this set consists of distinct families of designs. Since at the time of writing this document the work of *Roth et all (2017)* was still in progress, a new metric was developed that quantifies the quality of a layout. How this works is described in this section.

The metric is based on designer rationale captured by *Denucci (2012)*. He developed a Rationale Capture Tool (RCT) where designers could comment on automatically generated ship designs. These comments were structured and saved so that a resulting list of designer rationale emerged.

Although the captured rationale was obtained from naval architects discussing an offshore patrol vessel (OPV), many of the comments are also applicable for an MCMV. Therefore all rationale applicable to the MCMV was extracted from the total of 456 comments. Then all repeated comments were merged to just 10 comments. These, and their corresponding metrics are listed in Table 5.

Table 5: Applicable designer rationale for the MCMV from *Denucci (2012)*, including the metrics representing the rationale. Every metric should be minimized.

#	Designer Rationale	Reason	Metric
1	The length of fuel piping must be minimized	Survivability/Cost	Sum distances between all tanks and generators
2	Shaft length should be minimized	Space/Weight/Cost	Distance between propulsor and propulsion room
3	High ranked officer accom[13] should be close to the bridge	Operability	Max. distance between high ranked officer accoms and bridge
4	accom shouldn't be near the bow	High accelerations	Negative min. distance between accoms and bow
5	High ranked officer accom shouldn't be below dcd	Survivability	Count number of high ranked officer accom below dcd
6	Drystores should be close to the galley	Logistics	Max. distance between drystores and the galley
7	Bridge shouldn't be near the bow	High accelerations	Negative distance between bridge and bow
8	Davit shouldn't be too high above the waterline	Operability	Davit height minus the draft
9	accom should be grouped	Atmosphere	Within cluster sum of squares (WCSS) for k-means with k=2
10	accom shouldn't be close to heavy machinery	Noise	Negative min. distance between accom and generators, propulsion room and gun

[13] For each design the 14 accom blocks are first sorted on whether they are above dcd and are then sorted on their distance to the bridge. The first 4 accom blocks are then assigned to high ranked officers.

In order to see how effective these metrics are, the best ships in the total set regarding rational number 3, 7, 9, and 10 in Table 5 are displayed respectively in Figure 37:

- Design A shows that the pink accommodation blocks are grouped around the purple bridge. This ensures good access to the bridge for high ranked officers, enhancing operability of the ship.
- Design B has the bridge far from the bow. Therefore the bridge is closer to the center of rotation, minimizing seasickness for the captains.
- Design C has clusters of accommodation, which creates a nice atmosphere for the crew.
- Design D minimizes noise issues for the crew by placing all machinery in the aft of the ship, while all accommodation is near the bow. Furthermore the design is relatively long, which creates even more distance.

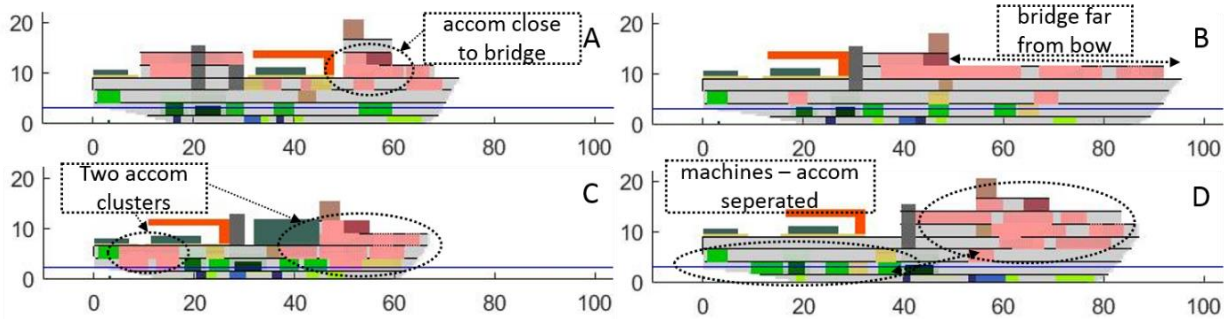


Figure 37: The best ships regarding the metrics defined in Table 5: High rank officers near the bridge (a), seasickness bridge (b), accom grouped (c), and noise (d).

The following step is to combine the ten metrics from Table 5 into one metric for the quality of the layout of the designs. First, since every metric has different values (for instance the first metric is typically in the order of tens, while the third metric is in the order of thousands) they are first standardized by taking their z-scores. Then for the sake of simplicity it is assumed in this thesis that every design comment is equally important, thus the quality of a layout is defined by the plain sum of these ten metrics without using a weight factor:

$$total\ quality\ of\ layout\ for\ a\ single\ design = \sum_{j=1}^{10} zscore(score\ metric_j)$$

Finally the 10% designs with the lowest total objective value are the designs with the best layout, and are therefore combined into a subset. This subset of 1715 designs is assumed to include only sensible designs, while remaining big enough to maintain diversity. It is investigated for clusters in the next section.

4.2.2 Apply method

The method applied to the resulting dataset with included designer rationale is illustrated in Figure 38, where each step will be elaborated in the following subsections.

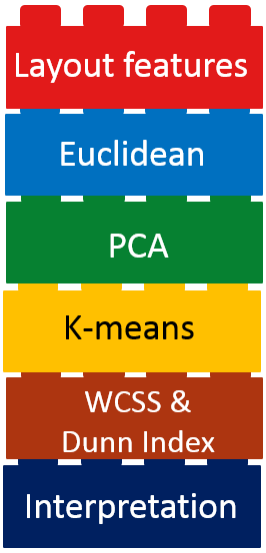


Figure 38: LEGO stack corresponding to the method applied in this section

4.2.2.1 Feature selection/creation

The dataset with included designer rationale is expected to consist of separated clusters regarding the layouts of the designs. Therefore the features regarding the positions of all objects in the layout are selected. In total the designs are packed with 43 objects. Since the designs are generated by 2.5D packing, x- and z-positions of all objects are selected, but y-positions are only deviating from the centerline for workshops and stores and are therefore omitted *van Oers and Hopman (2012)*. This results in a total of 86 selected features (or dimensions) which describe the designs.

Regarding normalization of the variables, all variables have the same unit of scale, and roughly operate on the same scale. Therefore it is not evident to use normalization. Additionally, chapter 4.1 showed that the discrete the tank top height had a big influence, while it only varied 0.5m. This was caused due to several systems which were always on the lowest deck, and thus their z-position only varied due to the tank top height. Normalizing these z-positions then makes the influence of the tank top height variation big. Therefore normalizing is omitted in this iteration.

4.2.2.2 Proximity measure

From the optional proximity measures in section 3.2:

- Mahalanobis distance is not used, since normalization is intentionally omitted.
- Cosine similarity is not used, since absolute differences matter (increasing the size of a design with a factor two does result in a different ship)
- Hamming distance is not used, since these features are real valued.

Therefore Euclidean distance is used.

4.2.2.3 Dimensionality reduction

The next step in the process is dimensionality reduction with PCA. An initial result is displayed in Figure 39, where the explained variance is plotted versus how many pc's are used (note that the pc's are sorted regarding the amount of variance they explain). The first pc does thus contain over 20% of the total variance. Furthermore it is interesting to see that 99% of the total variance is explained by using the first 33 pc's. This means that 86 – 33 = 53 dimensions can be discarded with very limited information loss. Figure 40 shows the data projected on the first 2 pc's. From Figure 39 it is clear that this plot contains about 29% of the total variance. Although there are no clear separate clusters visible, there is some structure present with regions that have a higher density. The presence of distinct clusters is further investigated in the next sections.

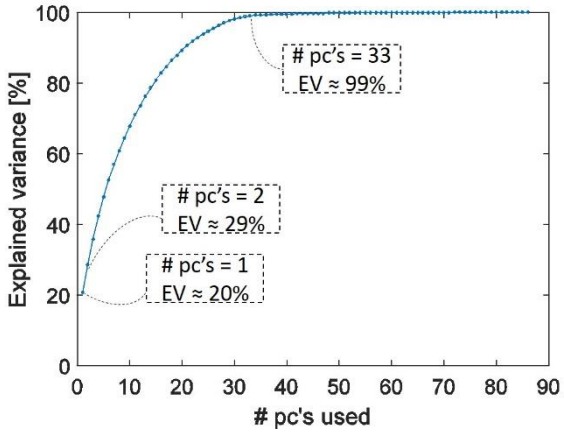


Figure 39: Explained variance vs. the number of pc's used for the MCMV dataset including designer rationale.

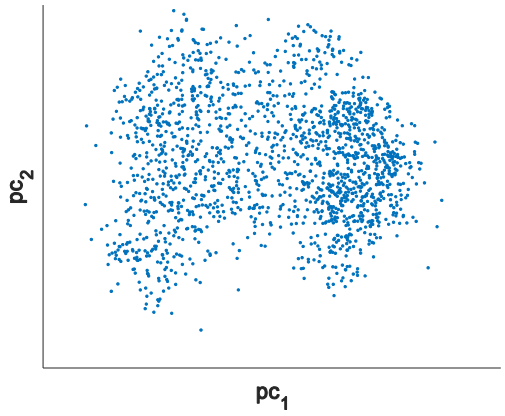


Figure 40: MCMV dataset including designer rationale plotted regarding its first 2 pc's.

4.2.2.4 Clustering

Next the k-means algorithm is applied to the reduced 33-dimensional dataset for various values of k. Since the 33-dimensional clusters are still hard to visualize in a figure they are validated in the next section. A projection of the clusters for k = 3 onto the first two pc's is shown in Figure 41.

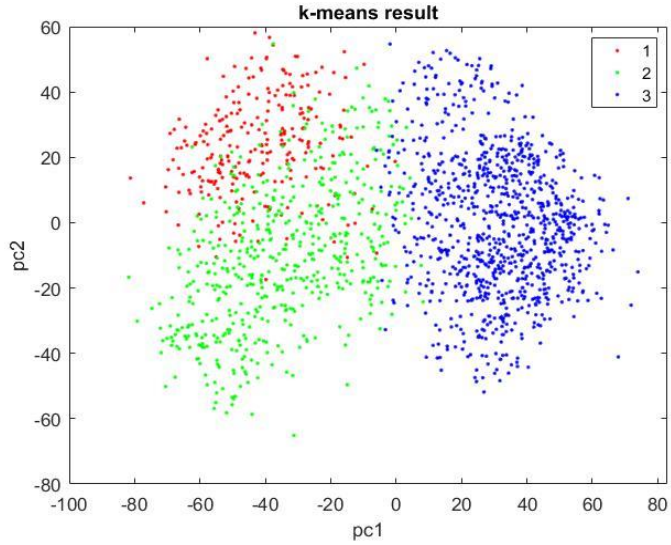


Figure 41: Data plotted in first two pc's clustered by the result of k-means for k=3

4.2.2.5 Validation

The Dunn indices calculated for the various cluster compositions from k-means are plotted in Figure 42. Instead of one value for k that corresponds to a high Dunn index, the Dunn index is approximately constant at a rather low value of about 0.18. This might indicate that there are no gaps in between the clusters. For further investigation, the Dunn index is calculated for 1000 random cluster compositions, Figure 43. The median of the resulting set equal 0.184, showing that there is roughly a 50% probability of a random cluster composition exceeding the cluster compositions from k-means. This means that there is no significant gap in between the clusters from k-means.

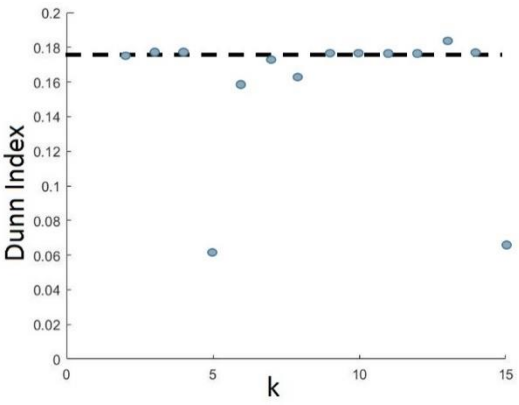


Figure 42: Validation of the results from k-means using the Dunn index

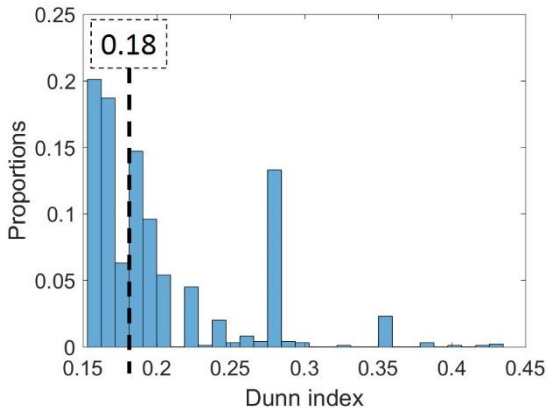


Figure 43: Histogram of Dunn index values for 1000 samples when the dataset is split into two groups by a random hyperplane[14].

[14] Hyperplane equidistant to two randomly selected data points.

4.2.2.6 Interpretation

Although the clusters are not separated by a significant gap, Figure 41 does show that the clusters correlate to certain regions with higher densities.

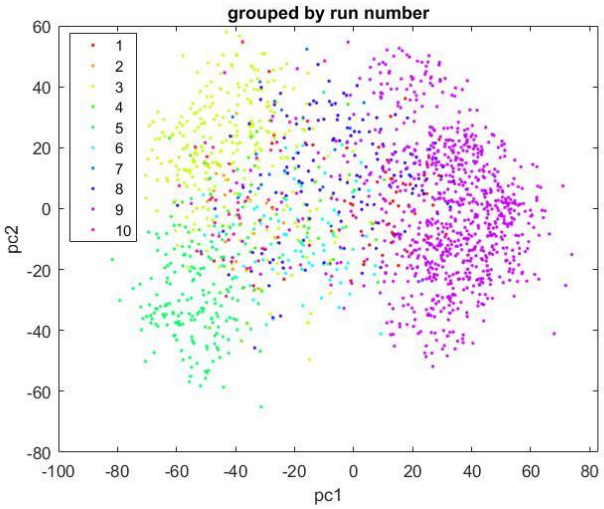


Figure 44: Data plotted in first two pc's clustered by the run number in which the data was generated.

The explanation for the visible structure lies in how the data was generated. For this specific case, the packing algorithm was run ten distinct times in order to compare it with another dataset. The results of these ten runs were combined to form the complete dataset *Duchateau (2016)*. Coloring the data based off this run number, as shown in Figure 44, and comparing this with the clusters found by k-means shows that the result is very similar:

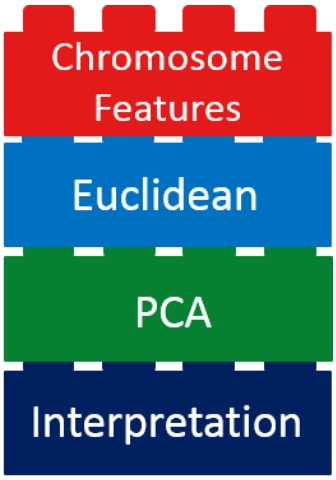
Cluster 1 corresponds to run number 3
Cluster 2 corresponds to run number 5 (and 4, 6, 8 and 10 which are smaller)
Cluster 3 corresponds to run number 9

This means that although it was expected to find structure in this dataset due to physical reason as motivated in Figure 36, the structure now seems to be fully dominated by the way the data was generated. On the other hand, some convergent behavior could be expected, since a genetic algorithm is used. The question is thus if the convergent behavior of the model is so apparent, that it dominates the physical structure. Or maybe there is no physical structure, and therefore the modelling aspects appear. Although the difference between the two might be delicate, the consequences are big, since this indicates whether the diversity in the dataset is sufficient to elucidate the expected physical clusters. Therefore further research is conducted in the next section, investigating how distinct the different run numbers really are regarding their chromosomal descriptions.

4.3 Chromosome analysis

It is expected that if there is a bias present due to the generation of the designs, that this bias will in particular be identified when looking at the chromosomes, since these are the values directly generated by the genetic algorithm. Therefore the set of chromosome features is further investigated for families in this chapter.

4.3.1. Iteration 1



In the first iteration PCA is applied to the chromosome data in order to get an initial idea which structure is present. The gene values are not normalized, since all genes already run between 0 and 1. In Figure 45, the data is projected on its first three pc's, and plotted for various rotations. The colors in the plots correspond to the ten distinct runs from the packing approach that were combined.

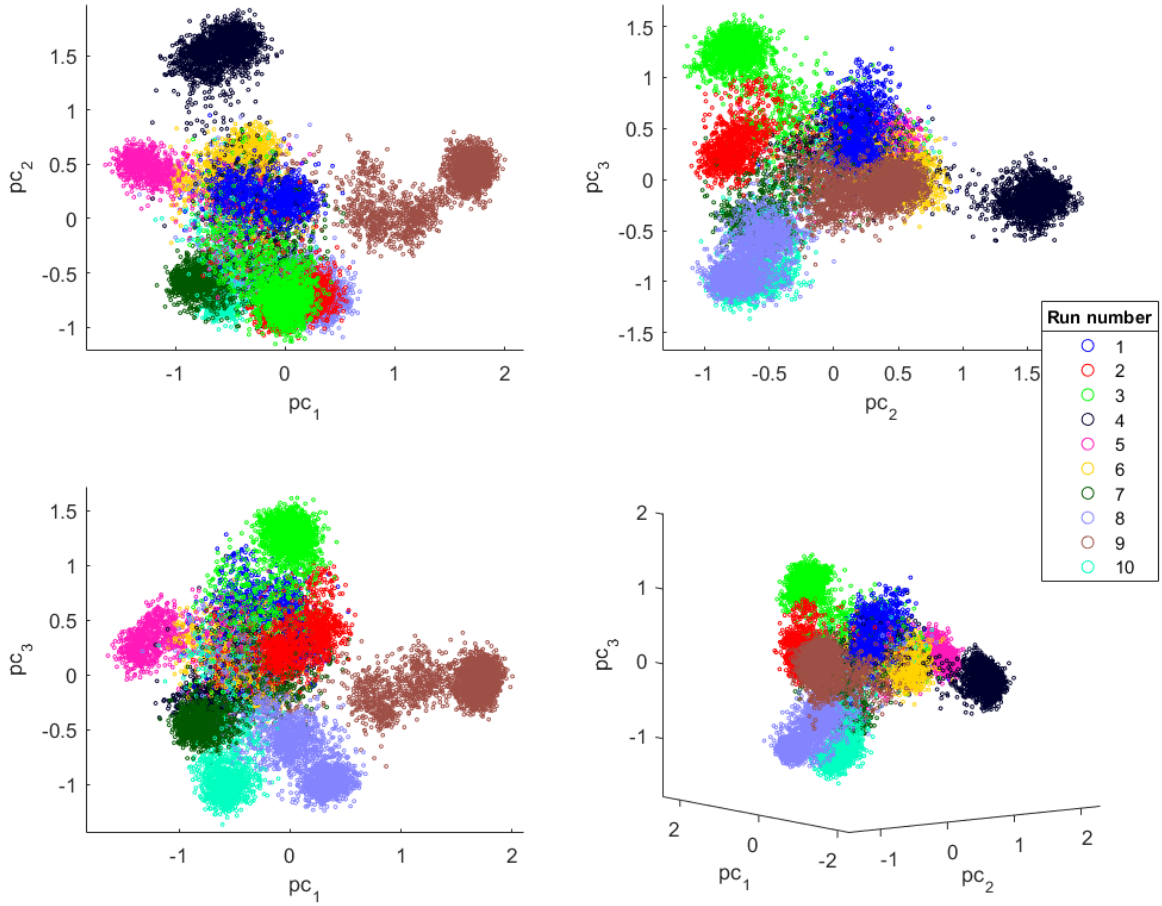
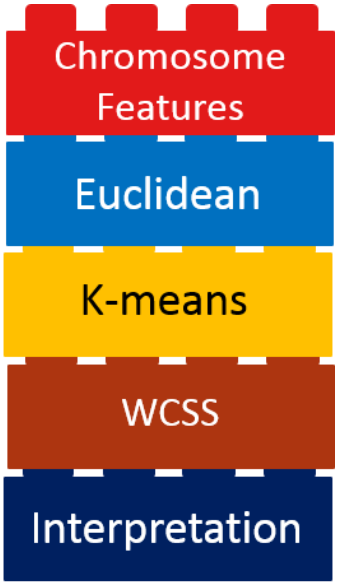


Figure 45: Chromosome data projected on its first three pc's, where colors correspond to the various run numbers of the packing approach.

Figure 45 clearly shows that every run explores its own part of the space. In fact, it looks like there is hardly any overlap between sets resulting from these various runs. These results thus seem to support the suspicion that the diversity for designs resulting from a single packing run is limited. However, these results only look at differences in a 3 dimensional subspace of the total "chromosome-space", so adding the other dimensions might set them apart even further. In order to test this, k-means is applied to the data in the following iteration.

4.3.2 Iteration 2



Given the results from the previous iteration, where the distinct runs were already clearly separated in 3D space, combined with the fact that k-means takes into account all dimensions, it is expected that it might be possible to reverse-engineer which designs were generated in the same run. This would then add meat to the suspicion of a lacking diversity from the packing approach. The chromosome data is therefore fed into the k-means algorithm for k=10, and a comparison of the resulting clusters with the various runs is depicted in Table 6.

Table 6: Comparison of the clusters resulting from k-means with the various runs.

		Run number									
		1	2	3	4	5	6	7	8	9	10
k-means clusters	10	1163	0	0	0	0	0	0	0	1	0
	1	0	1148	0	0	0	0	0	0	0	0
	9	0	0	1664	0	1	0	1	0	0	0
	7	0	0	0	1806	0	0	0	0	0	0
	3	0	0	0	0	857	0	0	0	0	0
	8	143	140	318	285	142	1073	247	170	375	176
	5	0	0	0	0	0	0	1139	0	0	0
	6	0	0	0	0	0	0	0	2057	0	0
	4	0	0	0	0	0	0	1	0	2955	0
	2	0	0	0	0	0	0	0	0	0	1292

Table 6 does show that the clusters resulting from k-means do align with the designs resulting from single runs. 88% of the designs are correctly classified to be generated in the same run, which is calculated by summing the diagonal values in Table 6 and dividing it by the total number of designs[15]. It is remarkable however, that almost all misclassifications are appearing in k-means cluster 8, and that this cluster contains a part of all ten runs. Plotting this cluster regarding the pc's shows that this cluster is in the middle of all data. This suggests that either all runs started somewhere and then converged to the region of cluster 8. Or when the data was generated, every run started at the same region, and then converged a different part of the design space. The ID numbers of the designs in cluster 8 then show that the latter is true. All

[15] The k-means clusters in Table 6 are rearranged such that this percentage is optimized.

runs were apparently initiated in the same way, and then converged to different parts of the design space.

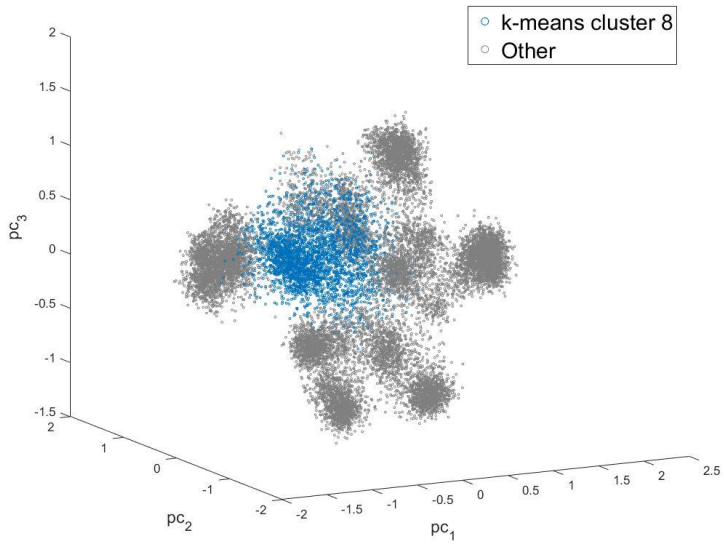


Figure 46: Chromosome data projected on its first three pc's, where cluster 8 resulting from the k-means algorithm is highlighted in blue.

The quest for families of ship designs resulting from the layout of the designs, resulted in a clue for lacking diversity in the packing approach. In this section the clue was first further investigated using PCA, leading to clearly visible distinct families that were caused by the run in which the designs were generated. This result was then strengthened by applying k-means, pointing out which designs were generated in the same run by just looking at the values of the chromosome.

5. Discussion & Future work

5.1 Continue the assessment of survivability of machine systems

The result that no multi-dimensional structure is present regarding the positions of machine systems, paves the way to define families regarding division of the individual features. For assessing survivability it is namely important in which compartments (between decks and bulkheads) these machine systems are located, and how these compartments relate to each other. A family could therefore be defined as all designs that have all machine systems within the same compartments. The exact location of a system within such a compartment is thus neglected.

In order to make an initial assessment of the feasibility of defining families this way, the number of families resulting from such a definition is estimated. Looking at the number of options for the z-positions of the various systems, the generator and propulsion rooms both remain on the lowest deck for all designs, thus having no variation. On the other hand, the z-positions of the main gun and the radar are distributed respectively over 4 and 3 decks. Regarding the x-positions, the average distance between two bulkheads for the MCMV is calculated to be 8.3m, while the spread of the x-position of the generator and propulsion room are respectively 65.5m and 10.3m. This means that they are approximately spread over respectively 8 and 2 longitudinal compartments. On the other hand, the radar and gun are not placed inside longitudinal compartments. Therefore the gun has 2 placing options (either on the bow or on the stern), and for the sake of simplicity, the x-position of the radar is assumed not to add variety. Calculating the total number of variations then results in roughly over 400 families.

Table 7: Calculation of the number of possible families

	With gun		Without gun		
	x	z	x	z	
Propulsor	1	1	1	1	
Prop. Room	2	1	2	1	
Gen. Room	8	1	8	1	
Radar	N/A	3	N/A	3	
Main gun	2	4	N/A	N/A	
Total options	**384**		**48**		**= 432**

This initial estimation shows that it is feasible to divide the set into about 400 families in order to assess the survivability of machine systems. However, the quantity is still rather larger, causing a fairly time consuming exercise to manually assess 400 representative designs. In fact, when the model is changed, and the packing algorithm is re-run, it could be that the process must be repeated.

5.2 Lacking diversity in Packing

The results showing the convergence of each packing run to its own part of the chromosome space is partly an inherent aspect of using a genetic algorithm as a search algorithm. But it is suspected that it is amplified by the following effect: The objective of the NSGA II algorithm

is to create compact designs (i.e. maximize packing density). When a design has a high packing density, it will therefore be higher graded than other designs. This causes the algorithm to keep trying to propagate this design. But in contrary this design is less likely to get a feasible child, since it is more difficult to pack the compartments in a denser design. The probability of having feasible children is then only higher if there are only minor changes applied, which causes convergence. Furthermore if the design does happen to get a feasible child, this is most probably the case due to increasing the size of the ship, which makes it of less quality then the original design.

This suspicion can be tested by investigating the family tree of a run, including the designs that have failed to meet the constraints. Since this information is not included in the data yet, this is up for future work.

If this suspicion is true, eliminating the relation between the probability of generating a feasible design, and the size of the envelope should then improve the diversity of the designs. This could for instance be obtained by making sure that if a design could be packed to a feasible solution based on its global parameters/envelope, ensuring a high probability that this solution will be found.

5.3 Reduced layout space

Applying PCA to the layout space showed that the layout space could be represented by only 33 pc's that contain 99% of the total information. At the moment, the chromosome feeding into packing uses 86 genes to represent the 86 variables of the 43 objects (x and z position for each object). The result from PCA suggests though, that this number of genes corresponding to the layout of the design could be reduced drastically (ideally to 33 components), while still containing almost all information. The advantage would be, that the chromosome for the MCMV would reduce to 53 genes (instead of 106), making it much more suitable to do a systematic variation of all genes to map the design space (instead/additive to variations with the genetic algorithm). Furthermore it could improve the packing approach to focus on feasible variations.

6. Conclusion

In this thesis the research objective was to:

Elucidate families of ship designs within the design set resulting from the packing approach, leading to new knowledge of the data at hand.

The application of a clustering method using techniques as PCA and k-means to the datasets resulting from the packing approach was proposed in order to achieve this objective. Before applying this method to datasets at hand, high peaks in the histograms from the cruise ship were identified, opposing the dataset of a MCMV. This motivated to use the MCMV dataset for the test cases. Then the method was applied to several test cases in order to evaluate its applicability:

- In a first test case, families are sought regarding the positions of the machine systems of an MCMV. The goal was to use these families in order to reduce the data by selecting representatives for assessing their survivability. Although no applicable division of the set was found, the method did reveal the present structure in the data. Even more important, the analysis made it plausible that no other structure is present in this data, thus generating new knowledge on the design space. Furthermore it paves the way for more straightforward definitions of families as discussed in section 5.1.

- In a next test case, the total layout from the MCMV was analyzed. The result seems to be dominated by the fact that the dataset was built by combining data from ten distinct runs of the packing approach. This result was amplified by analyzing the structure of the chromosome space. Although the new information did not reveal knowledge on the physical aspect of the designs, it did reveal knowledge on how modelling aspects cause structure in the data. It shows that despite the effort of using the NSGA II algorithm as a search algorithm by setting its mutation rate rather high *Duchateau (2016)*, the diversity for a single packing run is still limited. The lesson learned is twofold. It first shows that the resulting datasets from the packing approach should be treated with care, bearing in mind that the set is not as diverse as it might seem. Second it motivates to improve the packing approach, so that more diverse datasets will be created in the future.

The method revealed the underlying structure of the data in all test cases, which led to new knowledge. The research goal has thus been reached. Although no new knowledge is obtained on the physical aspects of the designs, the new knowledge that the results from the packing approach lack diversity is especially very helpful. Furthermore, it would have been difficult to gain these insights using other techniques. The clustering method allows to look at the data from a different point of view than ordinary plotting variables, and therefore creates new insights and hypotheses. The author is therefore convinced that its application will remain useful, revealing information about both the model as the relation between design and performance space.

7. Contributions

In this section is summarized what is contributed to research at the TU Delft:

- *Comparing designs stemming from the packing approach by calculating distance metrics.* In chapter 2, the calculation of distance metrics are proposed for comparing designs. The benefit of this method is that is both objective and fast, opposing other methods such as visual comparison.

- *Clustering methods are proposed, and applied to data resulting from the packing approach.* The quest to search for families of designs resulted in the proposition of applying clustering methods in chapter 3. In chapter 4 the techniques as PCA and k-means were applied to a test case of an MCMV, showing that new information was generated from this existing dataset.

- *Development of a new metric for the quality of layouts.* In order to make a selection of the best designs in the MCMV dataset in chapter 4, a new metric for the quality of the layouts was developed based on *DeNucci (2012).*

- *Visualizing convergent behavior of the packing approach.* The final results from chapter 4 show that each run of the packing approach converges to a certain region in the design space. This motivates new research aiming towards expanding the diversity of the designs resulting from the packing approach, where the techniques from this thesis remain applicable for visualizing the results.

Bibliography

ACKOFF, R.L. (1989), *From data to wisdom*, 16th Journal of Applied Systems Analysis, pp.3-9

ANDREWS, D.J. (1998), *A comprehensive methodology for the design of ships (and other complex systems)*, Proc. Mathematical, Physical and Engineering Sciences, vol 454, no. 1968, pp.187-211

ANDREWS, D.J. (2011), *Art and science in the design of physically large and complex systems*, Proc. R. Soc. A., pp.891-912

ANDREWS, D.J. (2013), *The true nature of ship concept design – and what it means for the future development of CASD*, COMPIT 2013, Cortona, pp.33-50

ARTHUR, D.; VASSILVITSKII, S. (2007), *K-means++: The advantages of careful seeding*, 18th ACM-SIAM Symposium on discrete algorithms

BREFORT, D.; SHIELDS, C.; SYPNIEWSKI, M.; GOODRUM, C.; SINGER, D.; HABBEN JANSEN A.; DUCHATEAU, E.; DROSTE, K.; JASPERS, T.J.M; ROTH, M.; HOPMAN, J.J.; KANA A.A.; PAWLING, R.; ANDREWS, D.; BROWN, A.; KARA, M.Y.; PARSONS, M. (2017), *An Architectural Framework for Distributed Naval Ship Systems*, Under Review in: Ocean Engineering

DENUCCI, T.W. (2012), *Capturing design: Improving conceptual design through the capture of design rationale*, PhD Thesis, Delft University of Technology

DING, C.H.Q.; HE, X. (2015), *k-means clustering via principal component analysis*, 21st Int. Conf. Machine Learning, Alberta, pp.29

DOERRY, N. (2004), *Measuring diversity in set-based design*, presented at ASNE Day 2015, Arlington, Virginia

DROSTE, K. (2016), *A new concept exploration method to support innovative cruise ship design*, Master Thesis, Delft University of Technology

DUCHATEAU, E.A.E.; OERS van, B.J.; HOPMAN, J.J. (2015), *Interactive steering of an optimisation-based ship synthesis model for concept exploration*, 12th IMDC, Tokyo

DUCHATEAU, E.A.E. (2016), *Interactive evolutionary concept exploration in preliminary ship design*, PhD Thesis, Delft University of Technology

FOX, E.; GUESTRIN, C. (2015), *Machine learning: clustering & retrieval*, University of Washington, Coursera, <https://www.coursera.org/learn/ml-clustering-and-retrieval>

HAMMING, R.W. (1950), *Error detecting and error correcting codes*, The Bell Systems Technical Journal, vol. 29, no. 2

HOPMAN, J.J. (2017), *Design of Complex Specials: Design Spiral & Systems Engineering*, TU Delft, lecture slides, course code: MT44035

JAIN, A.K. (2010), *Data clustering: 50 years beyond K-means*, Pattern Recognition Letters 31/8, pp.651-666

JASPERS, T.J.M.; KANA, A.A. (2017), *Elucidating families of ship designs using clustering algorithms*, 16th COMPIT, Cardiff, pp.474-485

KOHONEN, T. (1990), *The self-organizing map*, Proceedings of the IEEE, vol. 78, no. 9, pp. 1464-1480

LINSKER, R. (1989), *Self-organization in a perceptual network*, Computer 21/3, pp.105-117

LEEK, J.; PENG, R.D.; CAFFO, B. (2016), *Practical machine learning*, Hopkins University, Coursera, < https://www.coursera.org/learn/practical-machine-learning#>

LIU, H.; MOTODA, H. (2008), *Computational method of feature selection*, Chapman & Hall/CRC

MAATEN van der, L.; HINTON, G.E. (2008), *Visualizing high-dimensional data using t-SNE*, 9th JMLR, Tokyo

MEILĂ, M.; SHI, J. (2001), *A random walks view of spectral segmentation*, AISTATS

OERS van, B.J. (2011), *A packing approach for the early stage ship design of service vessels*, PhD Thesis, Delft University of Technology

OERS van, B.J.; HOPMAN, J.J. (2012), *Simpler and faster: A 2.5D packing-based approach for early stage ship design*, 11th IMDC, Glasgow

ROTH, M.J.; DROSTE, K.; KANA, A.A. (2017), *Analysis of general arrangements created by the TU Delft packing approach*, 16th COMPIT, Cardiff, pp.201-2011

ROWLEY, J. (2007), *The wisdom hierarchy: representations of the DIKW hierarchy*, J. Information Science 33/2, pp.163-180

SCHUTT, R.; O'NEIL, C. (2013), *Doing data science: Straight talk from the frontline*, O'Reilly Media

SILERYTE, R.; D'AQUILIO, A.; DI STEFANO, D.; YANG, D.; TURRIN, M. (2016), *Supporting exploration of design alternatives using multivariate analysis algorithms*, simAUD, pp.215-222

SINGER, D.J.; DOERRY, N.; BUCKLEY, M.E. (2009), *What is set-based design?*, ASNE, vol. 121, no. 4, pp.31-43

THEODORIDIS, S.; KOUTROUMBAS, K. (2009), *Pattern Recognition (4th ed.)*, Elsevier

VASUDEVAN, S. (2008), *Utility of the pareto-front approach in elucidating ship requirements during concept design*, PhD Thesis, University College London

Appendix A: COMPIT paper

Elucidating Families of Ship Designs using Clustering Algorithms

Ted Jaspers, TU Delft, Delft/Netherlands, T.J.M.Jaspers@student.tudelft.nl
Austin A. Kana, TU Delft, Delft/Netherlands, A.A.Kana@tudelft.nl

Abstract

This paper proposes a method to elucidate families of ship designs generated by the TU Delft packing approach using data clustering algorithms. The authors explore whether commonly used data science techniques can extract new information from the existing data. To test this hypothesis this paper applies data clustering algorithms to a test case of layouts of a Mine Counter Measures Vessel (MCMV) generated by the packing approach. Results look to improve the understanding of the multidimensional structure of the data, as well as to improve the comprehension and visualization of the complex interactions between the design and performance space.

1. Background

Early stage design is the initial phase in ship design where the balance between the different desired performances of the ship is explored. In ship design this process is often initiated by performing concept exploration, where various design solutions are explored in order to acquire knowledge about the interactions between design and performance space. Especially during concept exploration of complex ships, new requirements and/or new relationships between requirements can be elucidated. This results in an iterative process called requirement elucidation *Andrews (2013)*. On top of that, the traditional method of manually iterating through the design spiral is very time consuming. These aspects cause that in general only a small part of the design space is explored, which leads to an increased probability of converging to a suboptimal design, *Vasudevan (2008), Duchateau (2016)*. In order to explore the design space more extensively, the 'packing approach' was developed at the Delft University of Technology, which automatically generates tens of thousands of coarse feasible ship designs. By dividing this design space into ship design families, which share common design features and performance characteristics, the designer may then be able to better understand interactions between design and performance space. Think for instance of a ship where you can choose for either a long and narrow design corresponding to low resistance and low initial stability or short and beamy corresponding to high resistance and high initial stability. This paper explores the use of clustering algorithms to study the presence of ship design families stemming from the packing approach.

1.1 The packing approach

The packing approach is a tool that assists in enhancing the concept exploration process. The idea is to automatically generate a vast number of low level of detail feasible ship designs that cover a significant part of the design space. This is obtained by using a genetic algorithm on a parametric model of the desired ship, where all compartments in the ship are represented by building blocks. The designs are thus coarse, but detailed enough to calculate some performance measures (such as cost, speed, displacement, etc). Details on this approach can be found in *van Oers (2011), van Oers (2012)*. The resulting data set may consist of tens of thousands of designs, where each design has hundreds of design and performance attributes. This data is then structured and visualized so that information about the relation between design and performance space can be extracted *Rowley (2007)*. One visualization method applied to the packing approach is described in *Duchateau (2016)*. He proposed a method of displaying the data in matrix scatter plots, where numerical and architectural constraints could interactively be added. In Fig.1, L, B, GM and packing density are plotted, and the constraint added is that designs have deck 4 as damage control deck (dcd) instead of deck 5. Several results can be deducted from this figure. It shows for instance that the length and packing density are negatively correlated. The reason is that since a longer design has more space available, it has therefore more empty space to fit the same number of objects. Furthermore a higher dcd results in a lower GM, since objects (such as the main gun) should be above dcd, which raises the center of gravity.

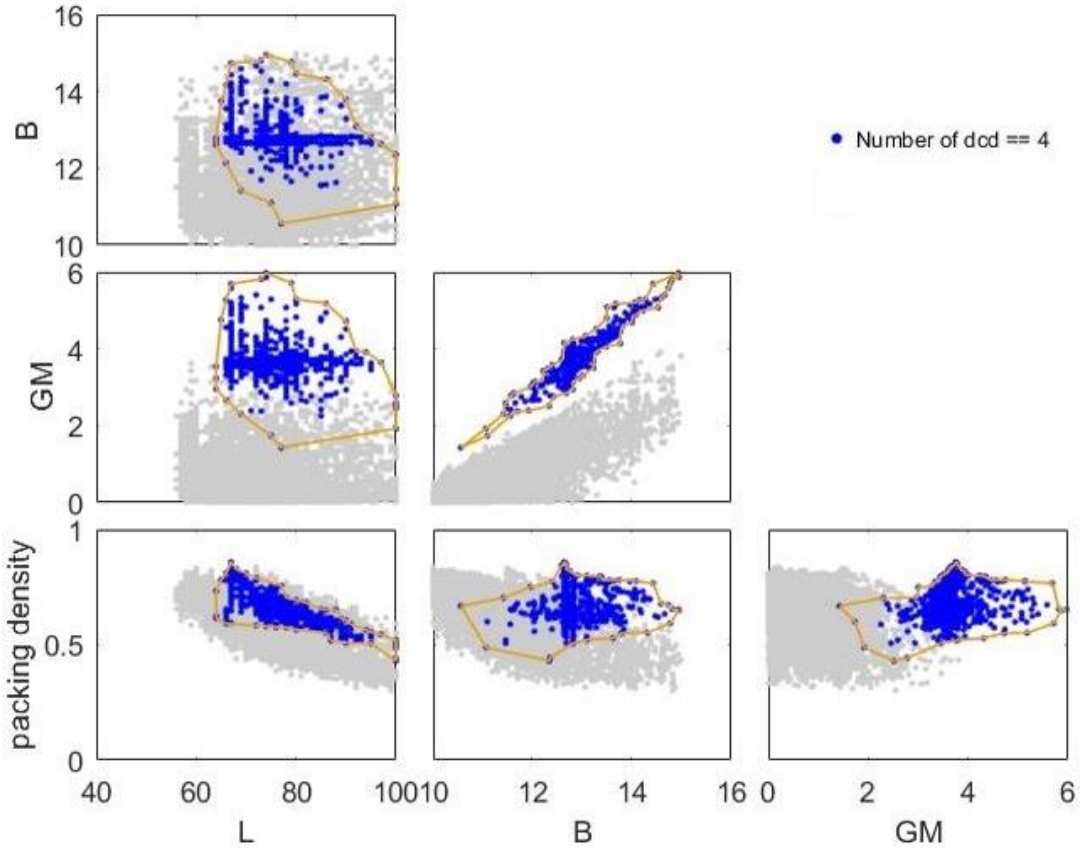


Fig.1: Visualization method of packing data as described by *Duchateau (2016)*. Various features are selected for plotting in matrix plots and constraints can interactively be applied.

Although the plots in Fig.1 contain a lot of information, they also raise new questions. Looking for instance at the clusters present in the middle plot, they are not separated in the plot in the lower left corner. So looking at the data from different directions determines whether the clusters are visible. It is therefore questioned whether there are other ways of looking at the data that reveal more structure.

1.2 Families of ship designs

In this paper families of ship designs are defined as being subsets of designs that share a clear similarity within design and performance attributes. These families should be clearly different when comparing designs between different subsets. An example is the middle plot in Fig.1, where two families are present. Another example is in *Droste (2016)*, where he defined different luxury levels for a cruise ship design. Examining the impact of these families in a performance space is shown in Fig.2. High luxury level causes a jump in both building costs and earning potential, creating two distinct families, clearly showing a tradeoff to be made. Identifying families of ship designs is thus very useful, especially when they correspond to certain regions of the performance space.

In contrast to these obvious families, which are defined by single discrete variables, families can be more complex. These complex families account for the inherent interaction between multiple design and performance features. These multi-dimensional families may be hard to identify and study using 2D plots. Looking for instance at the layout of a naval ship, there are numerous interactions between the compartments. Examples are: no machinery near accommodation due to noise and no accommodation near the bow due to seasickness. Thus, although the positions of objects are continuous variables, there might be discrete valid combinations of these variables, which results in clusters. A certain cluster might then for instance require a larger beam, corresponding to the part of the performance space with high resistance and high stability. But since the layout is depending on many features (such as x, y and z position of all compartments) these families are only detectable

using the combination of the features. An example of how this would look like from a data point of view is illustrated by the artificial dataset displayed in Fig.3. If we look at the data from the 2D plots (3a-c), there is no special structure present. But if we look at it in 3D space (3d), the data actually consists of two distinct families. These are exactly the type of structures that are sought in this paper, where it is hypothesized that they exist in higher than three dimensions.

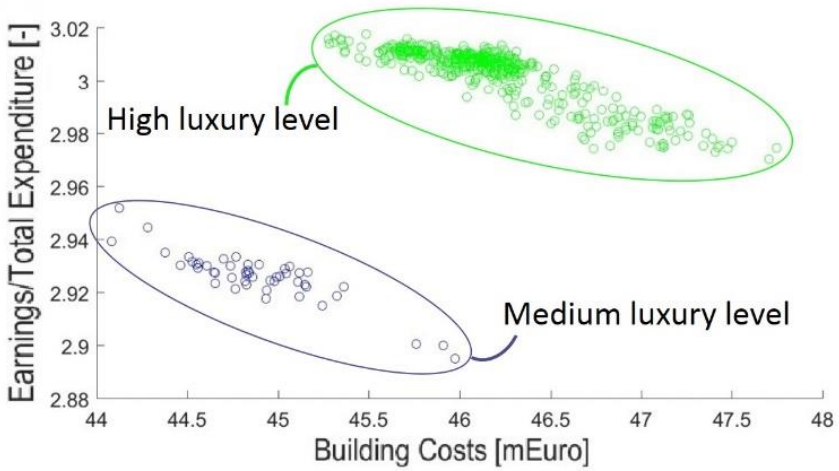


Fig.2: Various designs plotted in performance space, divided into families based on luxury level, *Droste (2016)*

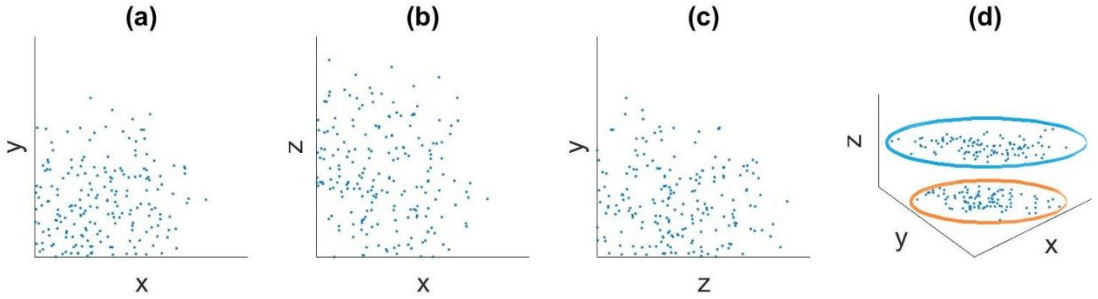


Fig.3: In this artificial dataset no clusters are detected by looking at 2d plots of (a), (b) and (c), whereas looking in 3d does reveal two clusters (d).

1.3. Clustering

In order to find these multidimensional families of ship designs, clustering algorithms from machine learning are proposed. These algorithms, as their name suggests, are devoted to find clustering structures in data. An example of their application is in companies as Facebook and Google where people are divided into clusters to achieve better assessment of which advertisement suits which person best *Schutt and O'Neil (2013)*. The analogy is that in this case the designs are divided into clusters to achieve better assessment of which design decisions suits which performance requirement best. The assumption is that the more distinct the clusters are, the more information they reveal about the relation between design and performance space.

A problem is however, that there is no clear notion of what 'being distinct' means. Studying for example the designs in Fig.4, there are various ways of comparing them. Looking at main dimensions designs A, B and C are similar, while design D is a bit longer. Whereas looking at the position of the working deck designs A and C have a working deck amidships, while at designs B and D it's positioned at the stern. Finally if assessing the main gun, only design A has one, while it is absent in the rest of the designs. These examples show that clustering is inherently a subjective science, as there is no single right or wrong way to cluster any given data *Theodoridis and Koutroumbas (2009)*. It is therefore important to investigate various sensible ways of clustering the set of designs.

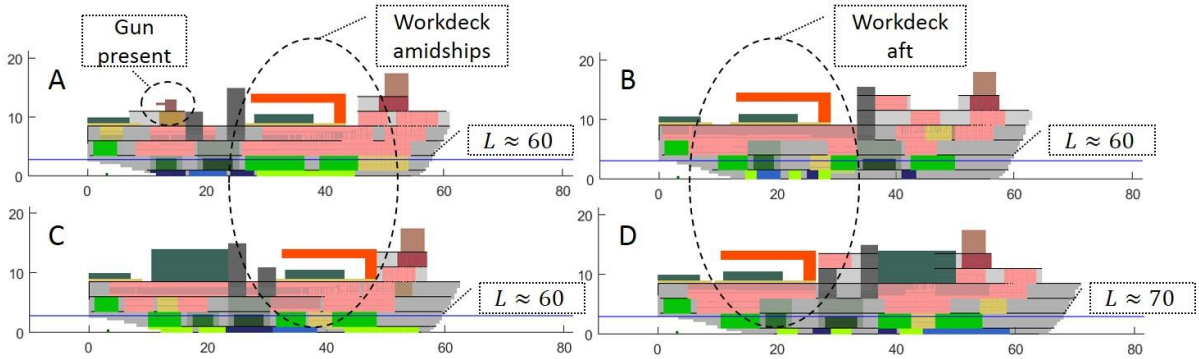


Fig.4: Four MCMV-designs resulting from the packing approach, where various differences and similarities are pointed out.

In this paper first the method for clustering is pointed out, including a more detailed description of the used techniques. Then this method is applied to find families of designs regarding the layouts of the MCMV.

2. Method

Due to the subjectivity of clustering, attention should be payed how to use the algorithms. The way you set up the problem partly determines which results you are going to find, and whether these results contain useful information. Clustering is a six step process (See *Theodoridis and Koutroumbas (2009)*):

1. Feature selection/creation: Select a set of features that are of interest. This may include all features, a subset of features, or new features. An example of a new feature is adding displacement, when L, B, T and c_B are available.
2. Proximity measure: Define how similarity between data points is measured by selecting a distance metric (such as Euclidean and Hamming distance).
3. Dimensionality reduction: Try to reduce the dimensionality of the data by using techniques such as Principal Component Analysis or Self-Organizing Maps. This improves the quality of the clustering algorithm as is shown by *Ding and He (2004)*. Furthermore the result can be used as initial investigation/visualization of the problem.
4. Clustering: If the remaining number of dimensions is more than three, apply a clustering algorithm, otherwise plotting data is possible. There are numerous clustering algorithms available.
5. Validation: Validate the clusters. This is not trivial since there are more than three dimensions, which makes it hard to visualize. Several metrics exist that indicate the quality of clusters.
6. Interpretation: When the result is valid, interpret it.

The specific algorithms used in this paper in steps 3, 4, and 5 are discussed in-depth more below.

2.1. Dimensionality reduction: Principal Component Analysis

This paper uses Principal Component Analysis (PCA) for reducing the dimensionality of the selected features. PCA is a valuable technique for exploratory analysis of high dimensional data. It rotates the original dataset in such a way that the first principal component (pc) corresponds to the direction with the highest variance, the second pc is orthogonal to the first pc and contains the second highest variance, and so on. A two-dimensional example is shown in Fig.5. This is useful for a number of reasons. Most important is that the amount of variance can be interpreted as being the amount of information *Linsker (1989)*. This reveals how PCA can be used for dimensionality reduction: Selecting and examining only those first couple of pc's that have the highest variance.

Since it is only possible to plot up to three dimensional data (higher dimensional plotting is technically possible (i.e. using colour and/or time), but the same argument holds.), a plot of the first three pc's will show you as much information as possible in one plot. On the other hand interpreting the content of the plot gets harder due to the complex values on its axis, since each pc is a linear combination of all input features. But the focus in this paper lies in identifying the multidimensional structure (clusters) in the data, which will still be visible in the plots. In fact, if there is a direction in space where clusters do show up, this direction has an increased probability of having a high variance, which makes it more likely to end up in the first three pc's *Ding and He (2004)*. This property is illustrated in Fig.6.

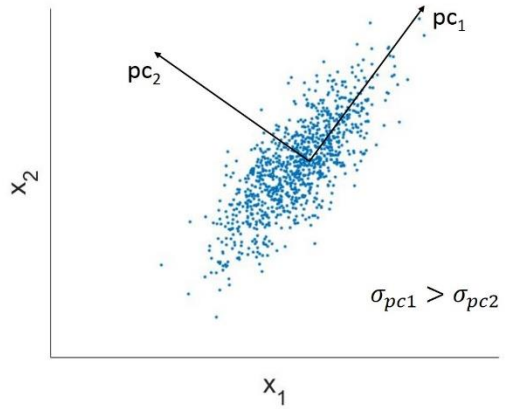


Fig.5: Illustration in 2D how PCA rotates the data. It makes it as "flat" as possible.

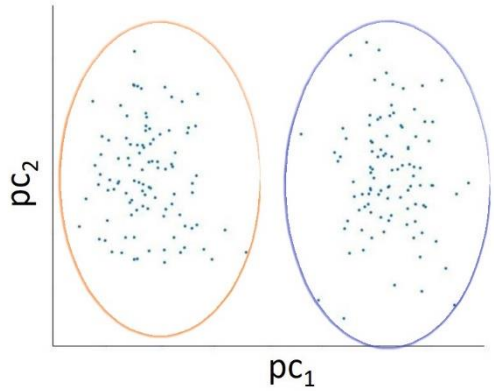


Fig.6: PCA applied to the data used in Fig.3. The first pc lies in the direction that reveals the clusters.

2.2. Clustering: k-means

K-means is a very popular clustering algorithm, mainly because of its elegancy and performance, *Jain (2010)*. The algorithm requires the amount of clusters, k, as input and uses the following steps:

- Initialize by selecting k distinct points $c_1, \cdots, c_k$ in space (Various initialization methods exist such as k-means++, which is used in this paper. The easiest is randomly selecting distinct positions as is used in the example of Fig.7.)
- Repeat until convergence:
  - Assign every data point to cluster i if it is closest to point c_i
  - Shift every c_i to the center of mass of the data belonging to cluster i

This process is illustrated in Fig.7. In Fig.7a the data itself and the random initialization of the centres is displayed. The first and second iterations are shown in respectively figures 7b and 7c, and finally convergence is reached in Fig.7d.

2.3. Validation: Dunn-index

There are a number of different metrics that give an indication of the quality of clusters. This paper seeks clusters based on physical attributes, which means that certain combinations of features result in infeasible designs. Thus, opposed to density clusters, there should be real gaps in-between the clusters. Therefore the Dunn-index is used, since it measures the size of the gap *Theodoridis and Koutroumbas (2009)*. It is defined as:

$$Dunn\ index = \min_{\forall i, \forall j \neq i} \left(\frac{d(C_i, C_j)}{\max_{\forall k}(diam(C_k))} \right)$$

Where $d(C_i, C_j)$ is the minimum distance between points from clusters C_i and C_j, and $diam(C_k)$ is the maximum distance between points from cluster C_k.

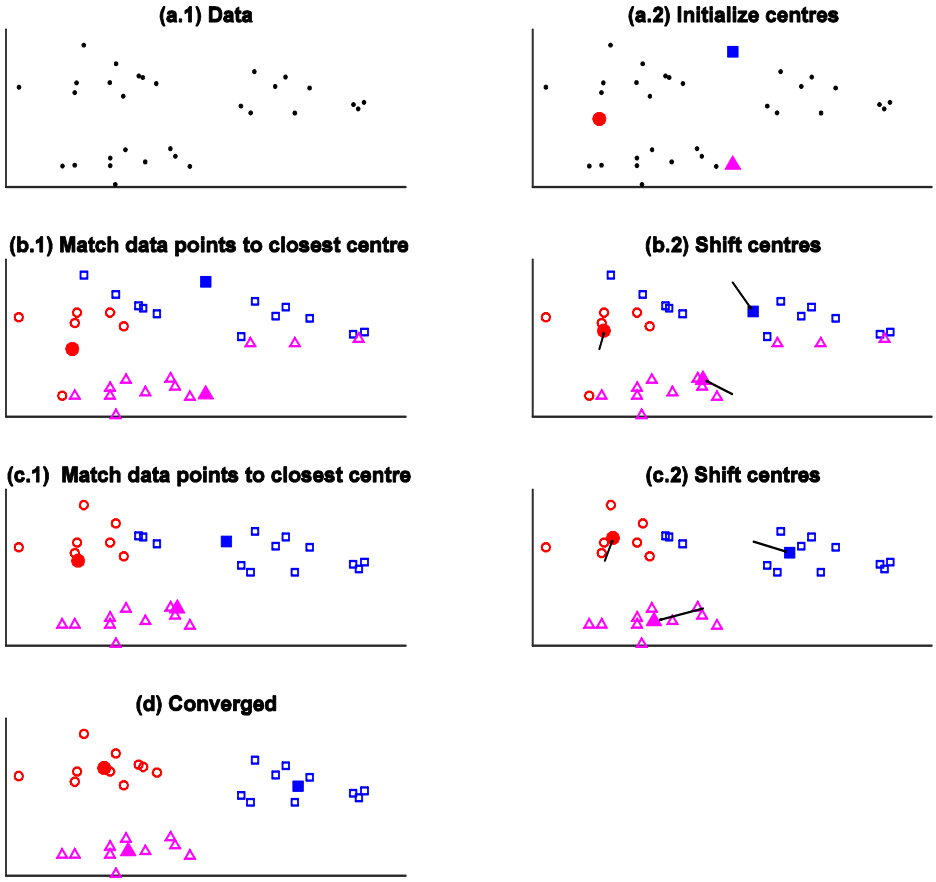


Fig.7: Illustration of how k-means converges on a 2D artificial dataset for k=3

To be more precise the Dunn-index is a measure that gives a lower bound for the distance between the clusters relative to the size of the clusters. A Dunn index of 1 or higher would therefore mean that the minimum distance between the clusters is higher than the diameter of the biggest cluster. In order to test whether the method in this section is able to elucidate families of designs from the data of the packing approach, it is applied to a test case in the next section.

3. Test Case: Mine-countermeasures vessel

The dataset of a MCMV as used in *Duchateau et al. (2015)* is used as test case. The set consists of over 17000 designs, with variations in global parameters (such as length, speed and range), optional objects (gun and Unmanned Surface Vessels (USV's)) and layout (position of the compartments and bulkheads). An example is displayed in Fig.8 including an explanation of its objects.

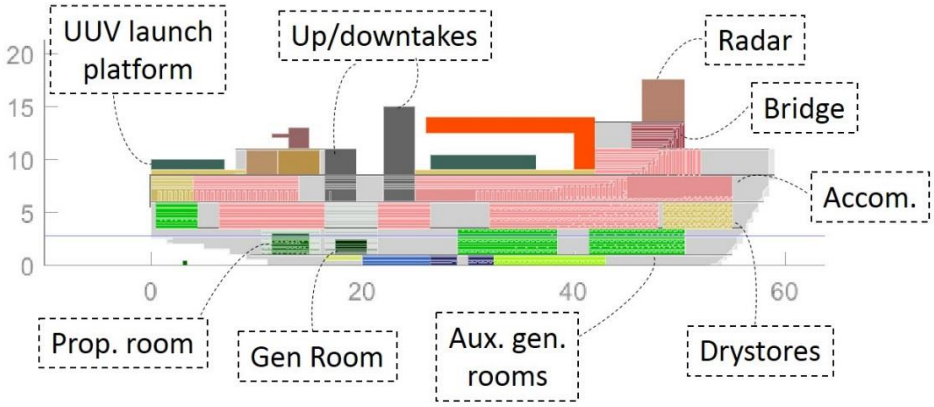


Fig.8: Example of a MCMV design from *Duchateau (2016)*

Since naval ships are complex designs that have many interactions between their compartments, the MCMV dataset is particularly interesting. The problem with this set is however that many of these interactions were omitted when initially developing the data set. This means that the compartments are more or less randomly stacked into the hull. Therefore, this section starts with selecting designs based on the quality of their layout. Then the proposed method is applied for finding multidimensional structures in the layouts of these selected designs. And finally, the results will be discussed.

3.1. Design selection based on designer rationale

The goal in this section is to select the 10% designs with the best layouts based on designer rationale. The layouts of the resulting subset of designs should then contain more structure than the remaining 90%. This enables verification of the hypothesis that this set consists of distinct families of designs.

In order to achieve this, first a metric is defined that quantifies if a design has a good layout. This is based on designer rationale captured by *Denucci (2012)*. He developed a Rationale Capture Tool (RCT) where designers could comment on automatically generated ship designs. These comments were structured and saved so that a resulting list of designer rationale emerged. All rationale applicable to the MCMV was extracted from that list and then quantified into a metric. The resulting ten comments and their corresponding metrics are listed in Table I.

Table I: Applicable designer rationale for the MCMV from *Denucci (2012)*, including the metrics representing the rationale. Every metric should be minimized.

#	Designer Rationale	Reason	Metric
1	The length of fuel piping must be minimized	Survivability/Cost	Sum distances between all tanks and generators
2	Shaft length should be minimized	Space/Weight/Cost	Distance between propulsor and propulsion room
3	High ranked officer accom[16] should be close to the bridge	Operability	Max. distance between high ranked officer accoms and bridge
4	accom shouldn't be near the bow	High accelerations	Negative min. distance between accoms and bow
5	High ranked officer accom shouldn't be below dcd	Survivability	Count number of high ranked officer accom below dcd
6	Drystores should be close to the galley	Logistics	Max. distance between drystores and the galley
7	Bridge shouldn't be near the	High accelerations	Negative distance between bridge and

[16] For each design the 14 accom blocks are first sorted on whether they are above dcd and are then sorted on their distance to the bridge. The first 4 accom blocks are then assigned to high ranked officers.

	bow		bow
8	Davit shouldn't be too high above the waterline	Operability	Davit height minus the draft
9	accom should be grouped	Atmosphere	Within cluster sum of squares (WCSS)[17] for k-means with k=2
10	accom shouldn't be close to heavy machinery	Noise	Negative min. distance between accom and generators, propulsion room and gun

The following step is to combine the ten metrics from Table I into one metric for the quality of the layout of the designs. First, since every metric has different values (the first metric is typically in the order of tens, while the third metric is in the order of thousands) they are first standardized by taking their z-scores[18]. Then for the sake of simplicity it is assumed in this paper that every design comment is equally important, thus the quality of a layout is defined by the plain sum of these ten metrics without using a weigh factor. Finally the 10% designs with the lowest total objective value are the designs with the best layout, and are therefore combined into a subset which is investigated for clusters in the next section.

3.2 Apply method

3.2.1 Feature selection/creation

The dataset with included designer rationale is expected to consist of separated clusters regarding the layouts of the designs. Therefore the features regarding the positions of all objects in the layout are selected. In total the designs are packed with 43 objects. Since the designs are generated by 2.5D packing, x- and z-positions of all objects are selected, but y-positions are only deviating from the centreline for workshops and stores and are therefore omitted *van Oers and Hopman (2012)*. This results in a total of 86 selected features (or dimensions) which describe the designs.

3.2.2 Proximity measure

There are multiple criteria for down selecting a proximity measure. Since in this case the data is in Cartesian coordinates, and consist of continuous variables, Euclidean distance has been chosen as an appropriate metric.

3.2.3 Dimensionality reduction

The next step in the process is dimensionality reduction with PCA. An initial result is displayed in Fig.9, where the explained variance is plotted versus how many pc's are used (note that the pc's are sorted regarding the amount of variance they explain). The first pc does thus contain over 20% of the total variance. Furthermore it is interesting to see that 99% of the total variance is explained by using the first 33 pc's. This means that 86 – 33 = 53 dimensions can be discarded with very limited information loss. Fig.10 shows the data plotted regarding the first 2 pc's. From Fig.9 it is clear that this plot contains about 29% of the total variance. As discussed in section 2.1.1, clusters are likely to show up in this plot. Although there are no clear separate clusters visible, there is some structure present with regions that have a higher density. The presence of distinct clusters is further investigated in the next sections.

[17] WCSS is equivalent to the sum of the Euclidean distances between every data point (accommodation) and its respective cluster center resulting from k-means.

[18] Taking the z-score of a data array sets its mean to zero and its standard deviation to 1 with the following transformation: $z_i = \frac{x_i - \mu_x}{\sigma_x}$ for all data points x_i.

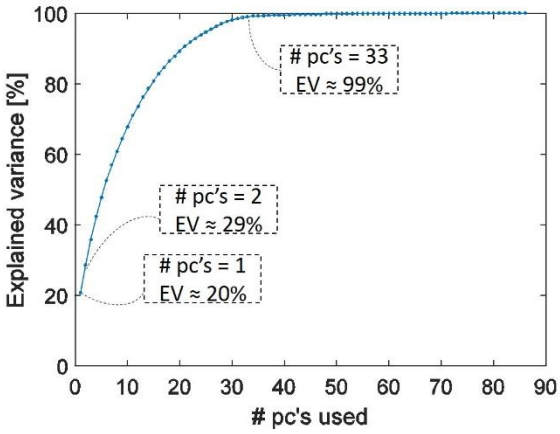


Fig.9: Explained variance vs. the number of pc's used for the MCMV dataset including designer rationale.

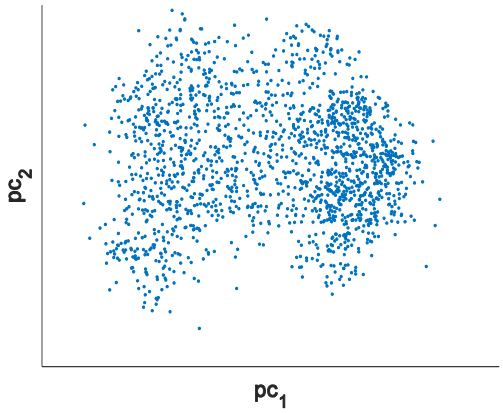


Fig.10: MCMV dataset including designer rationale plotted regarding its first 2 pc's.

3.2.4 Clustering

Next the k-means algorithm is applied to the reduced 33-dimensional dataset for various values of k. Since the 33-dimensional clusters are still hard to visualize in a figure they are validated in the next section. A projection of the clusters for k = 3 into the first two pc's is shown in Fig.13a.

3.2.5 Validation

The Dunn indices calculated for the various cluster compositions from k-means are plotted in Fig.11. Instead of one value for k that corresponds to a high Dunn index, the Dunn index is approximately constant at a rather low value of about 0.18. This might indicate that there are no gaps in between the clusters. For further investigation, the Dunn index is calculated for 1000 random cluster compositions, Fig.12. The median of the resulting set equal 0.184, showing that there is roughly a 50% probability of a random cluster composition exceeding the cluster compositions from k-means. This means that there is no significant gap in between the clusters from k-means.

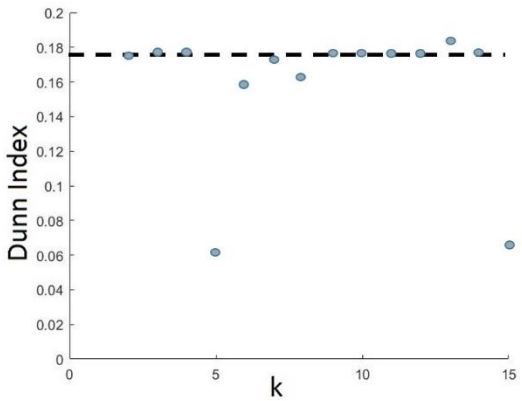


Fig.11: Validation of the results from k-means using the Dunn index

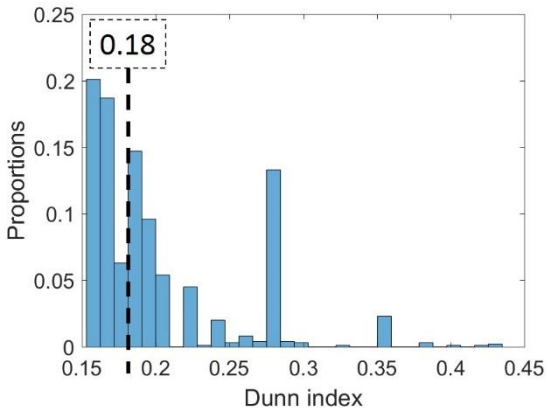


Fig.12: Histogram of Dunn index values for 1000 samples when the dataset is split into two groups by a random hyperplane[19].

3.2.6 Interpretation

[19] Hyperplane equidistant to two randomly selected data points.

Although the clusters are not separated by a significant gap, Fig.13a does show that the clusters correlate to certain regions with higher densities.

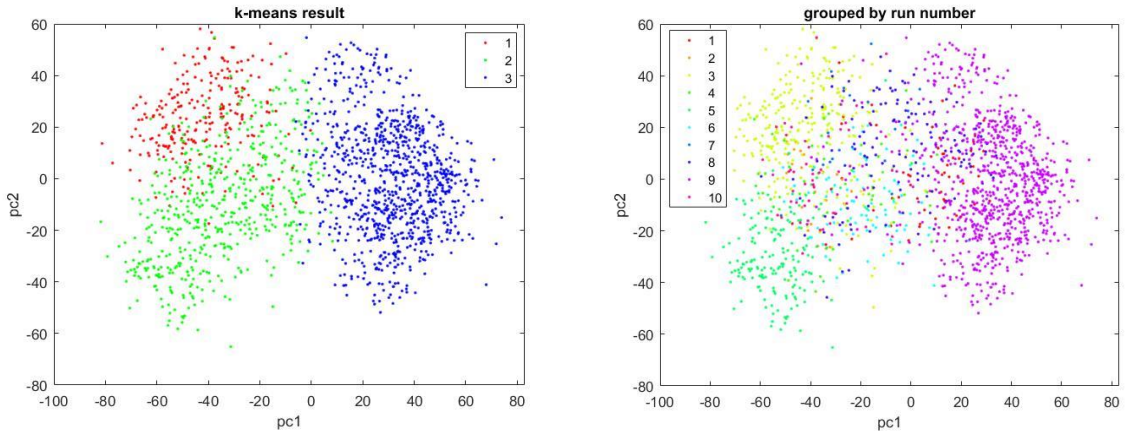


Fig.13: Data plotted in first two pc's clustered by: (a) the result of k-means for k=3 and (b) the run number in which the data was generated.

The explanation for the visible structure lies in how the data was generated. The packing algorithm was run ten distinct times in order to compare it with another dataset. The results of these ten runs were combined to form the complete dataset. Colouring the data based off this run number, as shown in Fig.13b, and comparing this with the clusters found by k-means shows that the result is very similar:

- Cluster 1 corresponds to run number 3
- Cluster 2 corresponds to run number 5 (and 4,6,8 and 10 which are smaller)
- Cluster 3 corresponds to run number 9

This result suggests that every run searches a particular part of the design space, since every run forms its own cluster based on layout. This notion is further investigated in the next section.

3.3. Design space

In the packing approach every design is parametrized by a chromosome which can then be adjusted by the genetic algorithm. The genes in the chromosome do thus form the design space. In order to further investigate the influence of run number in the dataset at hand, PCA is applied to this design space of the full design set, and the result is plotted in Fig.14. This static figure shows clearly that every run converges to a different part of the space, which is even better visible when the figure is rotated.

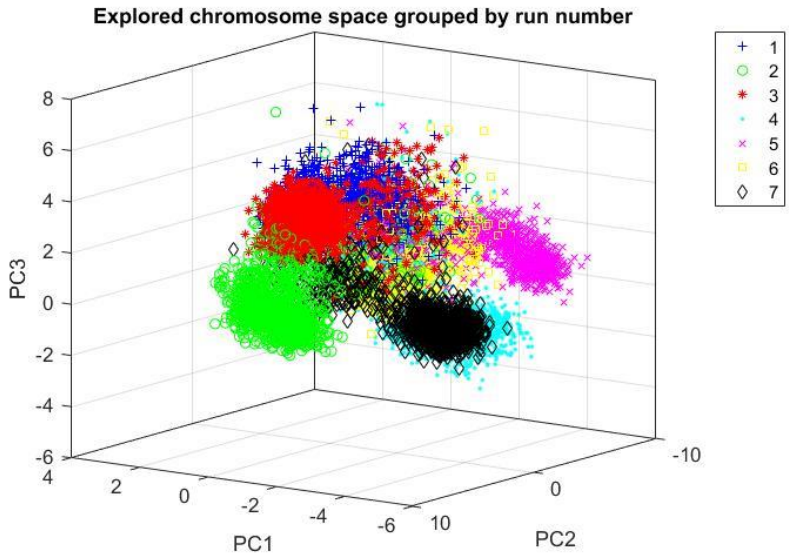


Fig.14: Input data plotted versus the first three pc's obtained from the design space. For visualization purposes only the first 7 out of 10 run numbers are included.

4. Discussion & Future work

Although part of the problem for the lacking diversity lies in using a genetic algorithm as a search algorithm, it is suspected that it is amplified by the following effect: The objective of the NSGA II algorithm is to create compact designs (i.e. maximize packing density). When a design has a high packing density, it will therefore be higher graded than other designs. This causes the algorithm to keep trying to propagate this design. But in contrary this design is less likely to get a feasible child, since it is more difficult to pack the compartments in a denser design. The probability of having feasible children is then only higher if there are only minor changes applied, which causes convergence. Furthermore if the design does happen to get a feasible child, this is most probably the case due to increasing the size of the ship, which makes it of less quality then the original design. This suspicion can be tested by investigating the family tree of a run, including the designs that have failed to meet the constraints. Since this information is not included in the data yet, this is up for future work.

5. Conclusion

In this paper clustering algorithms were used to search for families of ship designs generated by the TU Delft packing approach. Therefore a clustering method was applied to a test case of layouts from a MCMV. Unfortunately the results seem to be dominated by the fact that the dataset was built by combining data from ten distinct runs of the packing approach.

Although families of ship designs based on physical aspects of the designs were not clearly visible, the techniques used in this paper did reveal information on how the model generated the data. It shows that despite the effort of using the NSGA II algorithm as a search algorithm by setting its mutation rate rather high *Duchateau (2016)*, the diversity for a single packing run is still limited.

It would have been difficult to ascertain this behaviour using other techniques. The clustering method allows to look at the data from a different point of view than ordinary plotting variables, and therefore creates new insights and hypotheses. The authors are therefore convinced that its application will remain useful, revealing information about both the model as the relation between design and performance space.

Acknowledgements

We gratefully thank Bijan Ranjbarsahraei for supporting us by sharing his wisdom on the various techniques and methods from the field of data science in regular meetings.

References

ANDREWS, D.J. (2013), *The true nature of ship concept design – and what it means for the future development of CASD*, COMPIT 2013, Cortona, pp.33-50

DENUCCI, T.W. (2012), *Capturing design: Improving conceptual design through the capture of design rationale*, PhD Thesis, Delft University of Technology

DING, C.H.Q.; HE, X. (2004), *k-means clustering via principal component analysis*, 21st Int. Conf. Machine Learning, Alberta, pp.29

DROSTE, K. (2016), *A new concept exploration method to support innovative cruise ship design*, Master Thesis, Delft University of Technology

DUCHATEAU, E.A.E.; OERS van, B.J.; HOPMAN, J.J. (2015), *Interactive steering of an optimisation-based ship synthesis model for concept exploration*, 12th IMDC, Tokyo

DUCHATEAU, E.A.E. (2016), *Interactive evolutionary concept exploration in preliminary ship design*, PhD Thesis, Delft University of Technology

JAIN, A.K. (2010), *Data clustering: 50 years beyond K-means*, Pattern Recognition Letters 31/8, pp.651-666

LINSKER, R. (1989), *Self-organization in a perceptual network*, Computer 21/3, pp.105-117

OERS van, B.J. (2011), *A packing approach for the early stage ship design of service vessels*, PhD Thesis, Delft University of Technology

OERS van, B.J.; HOPMAN, J.J. (2012), *Simpler and faster: A 2.5D packing-based approach for early stage ship design*, 11th IMDC, Glasgow

ROWLEY, J. (2007), *The wisdom hierarchy: representations of the DIKW hierarchy*, J. Information Science 33/2, pp.163-180

SCHUTT, R.; O'NEIL, C. (2013), *Doing data science: Straight talk from the frontline*, O'Reilly Media

THEODORIDIS, S.; KOUTROUMBAS, K. (2009), *Pattern Recognition (4th ed.)*, Elsevier

VASUDEVAN, S. (2008), *Utility of the pareto-front approach in elucidating ship requirements during concept design*, PhD Thesis, University College London