Cost optimization of decommissioning offshore structures with the Pioneering Spirit using model order reduction and genetic algorithms

by

C.W. Vossers

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday January 25, 2019 at 10:30 AM.

Student number:	4169522	
Project duration:	April 23, 2018 – January 25, 2019	
Thesis committee:	Prof. dr. ir. K.A. Riska,	TU Delft
	Dr. E. Lourens,	TU Delft
	Ir. T. van der Horst,	Allseas Engineering
	Ir. F. Rodenburg,	Allseas Engineering

This thesis is confidential and cannot be made public until January 25, 2024.

Abstract

Over 450 oil platforms need to be removed from the North Sea in the next decade. Each of these platforms is custom designed for installation and service lifetime, but not for decommissioning. Allseas' Pioneering Spirit is the largest heavy lift vessel in the world, capable of lifting complete topsides in a single-lift operation. The vessel's lift system places up to eight lift points against the bottom of the platform and uses hydraulic cylinders to transfer the weight from the substructure to the vessel in a matter of seconds. But even with this vessel, removing an oil platform is technically challenging; the structural integrity needs to remain guaranteed, while minimizing the lift preparation cost as the structure will be scrapped. The goal of this thesis is to define a general approach that uses cost optimization for the platforms' lift preparation scope.

The costs of the platform's lift preparation scope consists mainly of two things; installation of lift points and reinforcement of the structure to avoid exceedance of their design resistance. In order to minimize the cost, the challenge is to find a lift configuration which minimizes the combined cost of the number, type and location of the lift points and the amount of reinforcement. A solution should not exceed constraints such as the allowable load per lift point, the number of lift points and the relative height difference between lift points. A lift configuration is statically undetermined, and the relative height difference between lift points is used to get different load distributions for the same configuration. In this way, forces are redistributed to strong points in the structure.

Detailed finite elements models are used to prove the structural integrity of a topsides during decommissioning. The number of degrees-of-freedom of such models is far too large for them to be used in an optimization algorithm. Model order reduction is used to condense the mass and stiffness matrix to the possible lift points, while keeping internal forces of interesting elements related to these points to check the structural integrity of a solution. The reduction allows for hundreds of lift configurations to be solved per minute.

The minimal costs for a lift configuration in the defined domain is found with the genetic algorithm. The optimization can include an excess amount of lift locations of different possible types. The location and load distribution over the lift points are used to define the cost function. The algorithm makes a selection that suits the constraints. The cost function gives the sum of the cost of the number and type of lift points and the number and type of structural failures.

This thesis presents a systematic approach for finding the optimal lift configuration for any offshore structure. The method is tested on a case study topsides by comparing the standard configuration with the optimized configuration. It is shown that by using the proposed method, the areas that need structural reinforcements are reduced from ten to zero and the amount of lift points is reduced from eight to seven, leading to a cost reduction of 27% on the offshore preparation scope.

The algorithm shows promising results for this topsides removal project. It is possible to include a new finite element model into the algorithm, which together with a generic cost function, makes sure the algorithm is conveniently applicable to other topsides removal projects.

Preface

In front of you, you have the master thesis '*Cost optimization of decommissioning offshore structures with the Pioneering Spirit using model order reduction and genetic algorithms*', written to obtain the degree of Master of Science in Offshore Engineering at Delft University of Technology. This thesis marks the end of a delightful and informative time as student in Delft.

This thesis was made possible by Allseas Engineering and I am grateful for their support. I hope this thesis will help the company with future removal projects and I am looking forward to be a part of the team responsible for cleaning up the North Sea.

I would like to express my gratitude to all the members of my graduation committee, Kaj Riska, Eliz-Mari Lourens, Tim van der Horst and Freek Rodenburg for supporting and guiding me throughout these nine months. Your helpful contributions were very much appreciated. Tim and Freek, I want to thank you in particular for always having time for me when I needed help.

Last but not least, I would like to thank my friends and family for supporting me throughout the process.

C.W. Vossers
Delft, January 2019

Contents

Abbreviations

CoG center of gravity.

DOF degrees of freedom.

DP dynamic positioning.

FE finite element.

FEM finite element method.

GA genetic algorithm.

MSF main support frame.

PS Pioneering Spirit.

PSO particle swarm optimization.

SLS serviceability limit state.

TLS topside lift system.

ULS ultimate limit state.

Nomenclature

C_v	Shear coefficient.
C_x	Elastic critical buckling coefficient.
C_m	Moment reduction factor.
$C_{rf,\boldsymbol{j}}$	Costs for reinforcements of a member.
C_{worst}	Highest value of a feasible individual in an entire population.
D	Outside diameter of a tubular member.
E	Young's modulus of elasticity.
F_{CoG}	CoG variance factor.
F_{SLS}	SLS load factor.
F_{ULSA}	ULS Condition A load factor.
F_{cr}	Critical stress.
F_{dyn}	Dynamic load factor.
F_v	Shear force.
F_{weight}	Weight contingency factor.
K	Effective length factor.
L	Length thickness of a member.
$L_{ULS,lift}$	Factored load during lift operation.
$L_{self-weight}$	Load of the topsides.
M	Bending moment.
M_n	Nominal flexural strength.
P_n	Penalty of a constraint violation.
V	Beam shear.
Z_e	Elastic section modulus.
Z_p	Plastic section modulus.
β	Random number according to the Laplace distribution.
$\boldsymbol{A}$	Inequality constraints matrix.
$\boldsymbol{B}$	Strain-displacement matrix.
$\boldsymbol{C}$	Stress tensor.
$\boldsymbol{F}$	Vector of forces applied to the system.
$\boldsymbol{F}^e$	Vector of forces and moments in an element.
$\boldsymbol{G_{ot}}$	Static transformation matrix.
$\boldsymbol{K}$	Stiffness matrix.
$\boldsymbol{M}$	Mass matrix.
$\boldsymbol{b}$	Vector containing the maximum value of each inequality constraint.
$\boldsymbol{g}$	Vector of gravitational acceleration.
$\boldsymbol{k}^e$	Element stiffness matrix.
$\boldsymbol{u}$	Vector of degrees of freedom.
$\boldsymbol{u}^e$	Vector of degrees of freedom of an element.
$\gamma_{R,b}$	Partial resistance factor for bending strength for tubular members.
$\gamma_{R,c}$	Partial resistance factor for axial compressive strength for tubular members.
$\gamma_{R,t}$	Partial resistance factor for axial tensile strength for tubular members.
$\gamma_{R,v}$	Partial resistance factor for shear strength for tubular members.
$\gamma_{R2,b}$	Partial resistance factor for bending strength for other members.
$\gamma_{R2,c}$	Partial resistance factor for axial compressive strength for other members.

$\gamma_{R2,t}$	Partial resistance factor for axial tensile strength for other members.
$\gamma_{R2,v}$	Partial resistance factor for shear strength for other members.
λ	Column slenderness parameter.
λ_{flange}	Factor by which the flange thickness needs to be increased.
$\mathbf{x}$	Possible lift configuration.
$\rho_{concrete}$	Density of concrete.
ρ_{steel}	Density of steel.
σ_c	Axial compressive stress.
σ_t	Axial tensile stress.
a_n	An integer variable –in this case whether a lift point is used or not.
b_n	A continuous variable –in this case the height of a lift point.
$c_{concrete}$	Cost of offshore concrete per kilo.
c_{steel}	Cost of offshore steel per kilo.
f_v	Shear strength.
f_y	Yield strength.
f_c	Representative axial compressive strength.
f_e	Euler buckling strength.
f_{xe}	Representative elastic local buckling strength.
f_{yc}	Representative local buckling strength.
n_{LP}	Amount of possible lift points.
$n_{constraints}$	Amount of constraints violations of an individual.
n_{elem}	Amount of elements in the finite element model.
$p_{crossover}$	The probability that crossover happens.
r	Uniform random number between 0 and 1.
r_g	Radius of gyration.
s	Random number according to the power distribution.
t	Wall thickness of a member.
$u_{\boldsymbol{j}}$	Utilization of a member.
x_i	Variable of an individual (parent) in a population.
x_{i+1}	Variable of an individual (child) in a population.

1 Introduction

Introducing the thesis, the research set-up will be discussed throughout the first chapter. First the focus will lie on the background of the thesis: the dismantling of topsides. Next the problem is defined, concerning difficulties in the process of finding an optimal lift configuration. Subsequently the research and its main objectives are presented, followed by the problem solving strategy. The introduction ends with a concise overlook of the structure of this thesis.

1.1. Background

In order to provide energy to the world, oil and gas companies have installed around 470 offshore platforms in the North Sea alone [22]. These platforms are not mass-produced, but each is custom designed and optimized for its specific project. When these platforms reach the end of their life, they have to be removed according to the OSPAR Decision 98/3 on the Disposal of Disused Offshore Installations [28]. Dumping or leaving the platform in place is prohibited.

From a technical perspective, removing them is challenging because most of these platforms were built without having a decommissioning plan available. The bigger platforms were built in small modules. Starting from a basic platform, living quarters and all kinds of technical equipment were added to the structure. As a result a complicated platform emerged. Removing the platform using reverse installation techniques would require a great amount of (offshore) work on the platform, as every single module would need a reinstallment of lifting frames [29] and structural integrity checks. These structural integrity checks potentially could lead to more reinforcements in order to guarantee the structural integrity of the topsides. Furthermore, as every single module can only be removed under favorable weather conditions, the time needed to remove the complete topsides is highly dependant on the amount of calm weather windows.

Another approach to remove platforms is to lift the complete topsides in a single-lift by using a specially designed vessel. The biggest advantage of a single-lift removal in comparison with a multi-lift removal is the reduction of time needed to remove the complete topsides. Instead of removing the modules one after another, the complete topsides is removed within one lift. As an alternative to spending months, especially when the heavy-lift vessel has to wait for favourable weather, the removal operation itself can be done within days and is thus preferable over a multi-lift operation.

However, a big challenge of a one-piece removal is keeping the topsides intact during the lift operation. Platform strengthening is needed in order to guarantee the structural integrity of the topsides. Moreover, beams in the topsides which undergo a higher load than their design resistance, need reinforcements in order to stay intact. New beams and plates are added to the topsides to increase the strength, which has a significant cost. Another challenge of an one-piece removal is keeping the lift points intact. For a single lift operation, lift points have to be constructed on the topsides. The special designed vessel uses these lift points to lift the topsides from its jacket. Strengthening can be required to make sure the structural integrity remains at these lift points. As it is always a challenge to do work offshore, adding reinforcements to the topsides and constructing lift points takes a great investment.

To reduce the amount of investments needed for a single-lift operation, an optimization of the force distribution within the topsides can be done. The complete weight of the topsides is distributed through the platform to the lift points. A new force distribution in the topsides can be achieved by either changing the locations of the lift points or adjusting the heights of the lift points. The combination of the lift locations and the height of each lift point is called a lift configuration. By changing this lift configuration, the force distribution in a topsides is modified and subsequently can lead to a reduction in the amount of members which have a higher load than their design resistance. In this way the amount of reinforcements done on the platform can be reduced. Choosing a favorable lift configuration hence has a great influence on the costs of a removal operation, as reducing the amount of lift points and the amount of reinforcements lowers the costs.

Allseas is a Dutch-based offshore company which has a great experience in the oil and gas industry. In 2016 Allseas commenced a special designed vessel for single-lift operations, Pioneering Spirit (PS), into operation. PS uses a dynamic positioning system, which makes it possible to sail around a topsides. The twin-hulled vessel is 382 m long and 124 m wide. At the bow is a lifting system is installed, that enables Pioneering Spirit to remove or install entire topsides up to 48,000 tons in a single lift by using up to eight sets of horizontal lifting beams.

1.2. Topsides lift system

Pioneering Spirit lifts topsides with the topside lift system (TLS), as is shown in fig. 1.1. It uses its u-shaped bows to sail around a platform and lift topsides up to 48,000 tons. On both bows four forklift units are installed, each of which lifts a part of the weight of a topsides. The forklift units are connected to the platform at different lift points, which are used to lift the topsides.

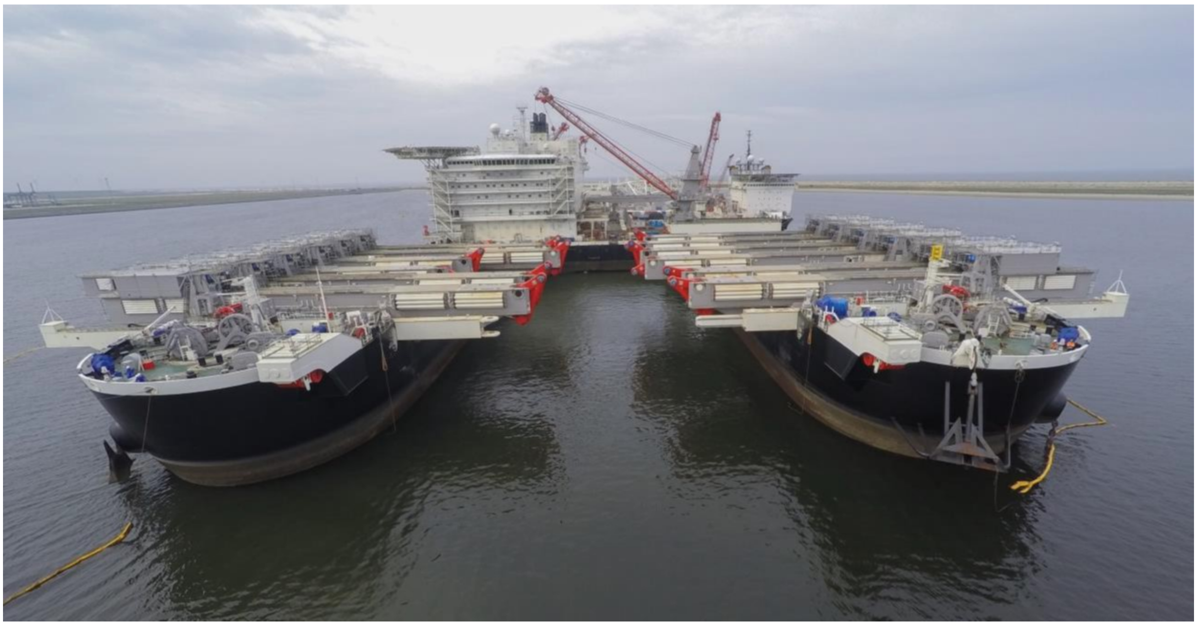


Figure 1.1: The topsides lift system of Pioneering Spirit, with four forklift units on each side

Figure 1.2 shows a detailed view of a forklift unit. The right side of the picture shows two lifting beams, which are connected with levers (shown in red) to a yoke foundation (shown in green). These levers are used to adjust the height of a lift point, as explained in section 1.3.1. The small blue pin on the left is a load pin, which is in essence a strain gauge. The yoke foundation is connected to the yoke itself (shown in brown), which is the connection from the TLS to the top sides.

In the lifting beams two cylinders are present (fig. 1.3), which make it possible to adjust the height of a yoke. The small cylinder is used for motion compensation, while the big cylinder is used for lifting the topsides itself. Before lifting a topsides, oil pressure is built up to the desired pressure, and held back with a valve. When it is time to lift a topsides, these valves are opened, in order to lift the complete topsides within seconds.

Before lifting a topsides, a lift configuration is designed as explained in section 1.3.1, which contains a certain reaction force at each lift point. When a topsides is lifted, the reaction force at each lift point is immediately determined using the load pin and the measured pressure in the lift cylinder. If the reaction force in a lift point is higher than anticipated, it is possible to relieve some oil pressure, in order to obtain the desired reaction force. Similar, when the reaction force in a lift point is lower than the designed reaction force, the

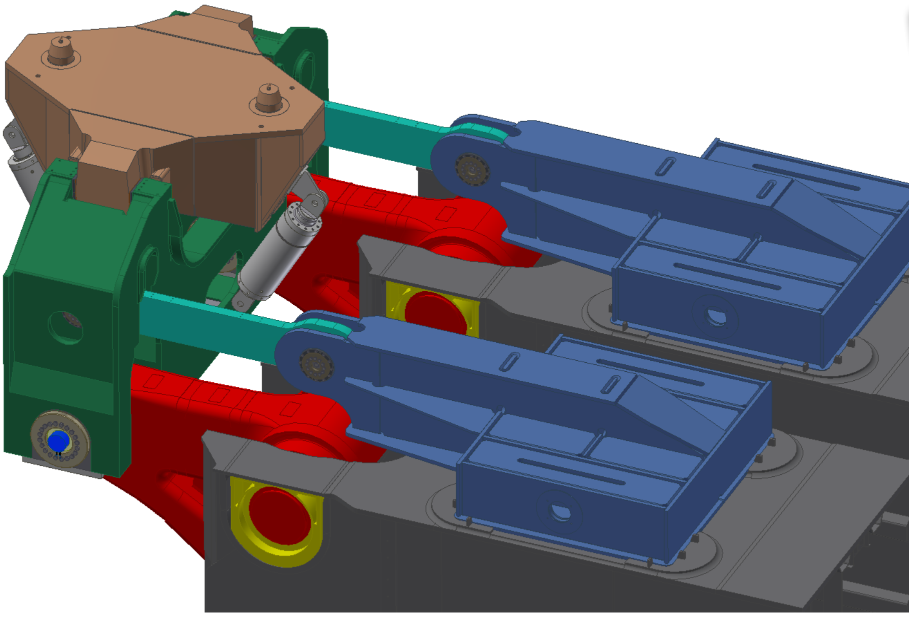


Figure 1.2: A detailed view of a forklift unit, showing the yoke foundation and load pin

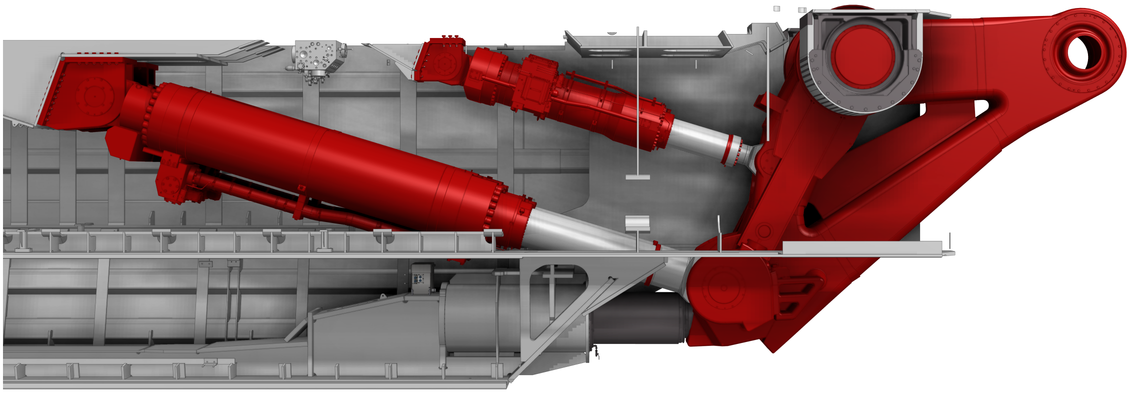


Figure 1.3: Exploded view of a lifting beam, showing the lift cylinder and the motion compensation cylinder

pressure in the lift cylinder is increased to raise the yoke, and consequently increasing the reaction force. In this way it is possible to actually achieve the desired lift configuration.

1.3. Problem formulation

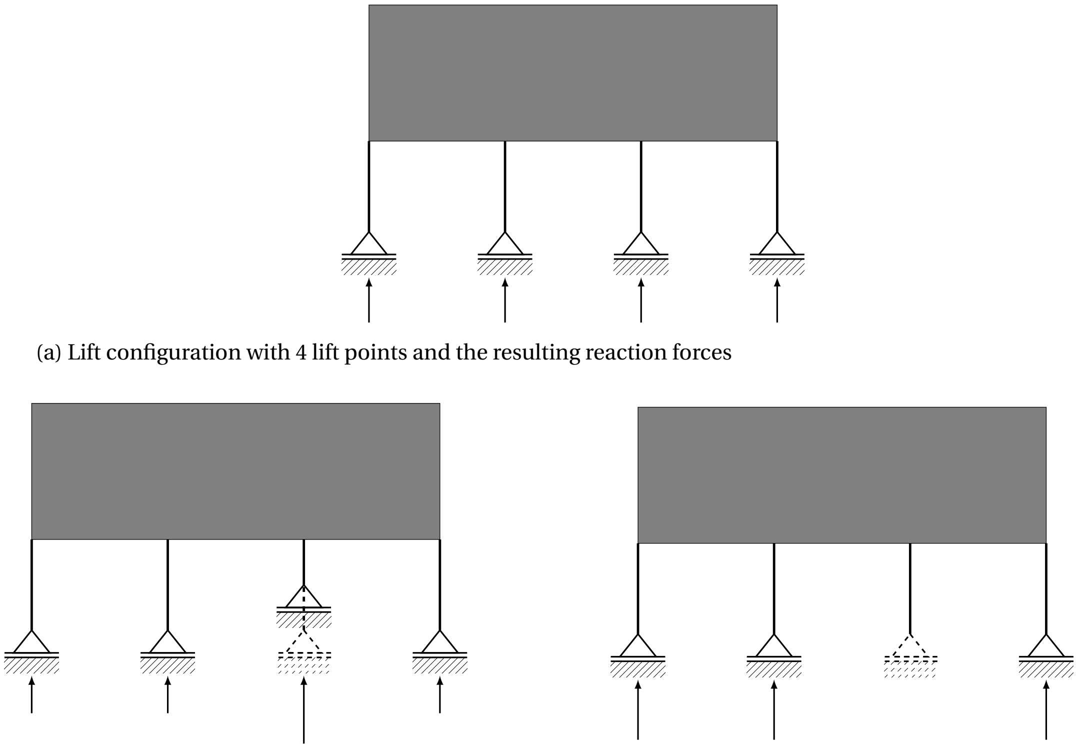
1.3.1. Problem definition

A schematic drawing of a topsides can be found in fig. 1.4a. The topsides is represented by the grey box and the vertical lines in the figure are the legs of the topsides. On each leg a lift point is constructed (represented by a floating bearing). Each lift point can have a different reaction force, determined by the structural configuration of the topsides, and the mass distribution of the topsides. In this example the reaction forces are distributed equally, as shown by the arrows underneath each lift point.

Because the mass distribution and structural configuration of a topsides is constant, the distribution of reaction forces can be changed by adjusting the height of a lift point. The location of a lift point on the leg stays on the exact same spot, and the length of the legs remains unchanged. By moving the lift point up, that lift point attracts a bigger part of the reaction force. This causes a reduction in the reaction forces of the other lift points, while the sum of reaction forces remains the same. This is shown schematically in fig. 1.4b, where the reaction force of the moved lift point has become higher, while the other reaction forces are lowered.

As shown in fig. 1.4c, it is also possible to lift the topsides with three lift points. While removing a lift point, the total reaction force has to stay the same, as the mass of the topsides does not change. Hence with removing a lift point, the reaction forces in the other lift points are increased.

In this example there were only four possible lift locations (per side), while in reality for most topsides it is possible to identify more than four lift locations. As PS has only four pairs of lifting beams per side, a maximum of four lift points is used per side. In addition, a minimum of two lift points per side is needed to guarantee a stable equilibrium. Those facts make it possible to use different combinations of lift points as lift configuration. Furthermore, as explained earlier, even two lift configurations which use the same lift points, can have a different distribution of reaction forces, by adjusting the heights of lift points.



(a) Lift configuration with 4 lift points and the resulting reaction forces

(b) The influence of adjusting the height of a lift point to the force distribution at the lift points

(c) The influence of removing a lift point to the force distribution of the lift points

Figure 1.4: Schematic drawing of a topsides (in grey), showing different possible lift configurations

The strength in a topsides is provided by a support structure, composed of different members. The reaction forces at the lift point are distributed into the topsides through this truss structure, and in this way the members are loaded. By adjusting the height of a lift point, the reaction forces in each lift point are changed (fig. 1.4b). Consequently, the loading in the members is changed as well. For some members the loading becomes lower, while for other members the loading is increased. If a lift point is added or removed, as shown in fig. 1.4c, the complete force distribution in the truss structure changes.

Topsides are not designed for a single-lift removal, which causes the loads in several members to exceed the design resistance of those members. As result, those members would fail if nothing would be done. To prevent failure, reinforcements are done on those members. However, as explained before, it is possible to adjust the loading in the truss structure by changing the lift configuration. Accordingly, it is possible to minimize the amount of members which need reinforcement.

The objective of this thesis is designing an approach, which makes it possible to find an optimal lift configuration for any topsides. An optimal lift configuration minimizes the amount of reinforcements, while at the same time the amount of lift points is kept to a minimum, as the construction of a lift point is costly as well.

1.3.2. Current versus proposed approach of designing a lift configuration

Currently, for designing a lift configuration, a finite element (FE) model is used. For some projects this FE model is supplied by the client, while for other projects a new model needs to be created. This is done by reproducing a 3D-model from technical drawings of the topsides. The properties of each member, stiffness of the complete structure and the mass distribution of a topsides should be modeled as close to reality as possible. This 3D-model is imported to FEMAP, a FE program, which is used for running simulations and adding boundary conditions. This makes it possible to model the structural behavior of the topsides.

Next, a project is created in SDC verifier. SDC verifier is a software program, which is used for verifying the strength of a structure. In a FE model, beams are often made up from different smaller elements. SDC finds the complete buckling length of members, as well as different parameters for each element, such as plastic and elastic section moduli, the bending strength and the yield strength. Furthermore, SDC verifier calculates strength utilizations, and hence determines whether a member fails or not.

After that, a heavy lift engineer, which is responsible for designing the single-lift operation, proposes a possible configuration, which contains the locations of the used lift points. Figure 1.5 shows that this lift configuration, together with the FE model and the SDC project, is checked for feasibility. This means that, for feasible lift configurations, the reaction forces are lower than the lifting capacity of the lifting beams, and that the amount of reinforcements stays within reasonable limits. If a lift configuration is feasible, the lift configuration is used. If the lift configuration is unfeasible, the heavy lift engineer has to come up with a new lift configuration. This can be done by either defining new lift points, adjusting the heights of lift points, or a combination of those two. This new lift configuration is again checked for feasibility, and this process repeats until a feasible lift configuration is found.

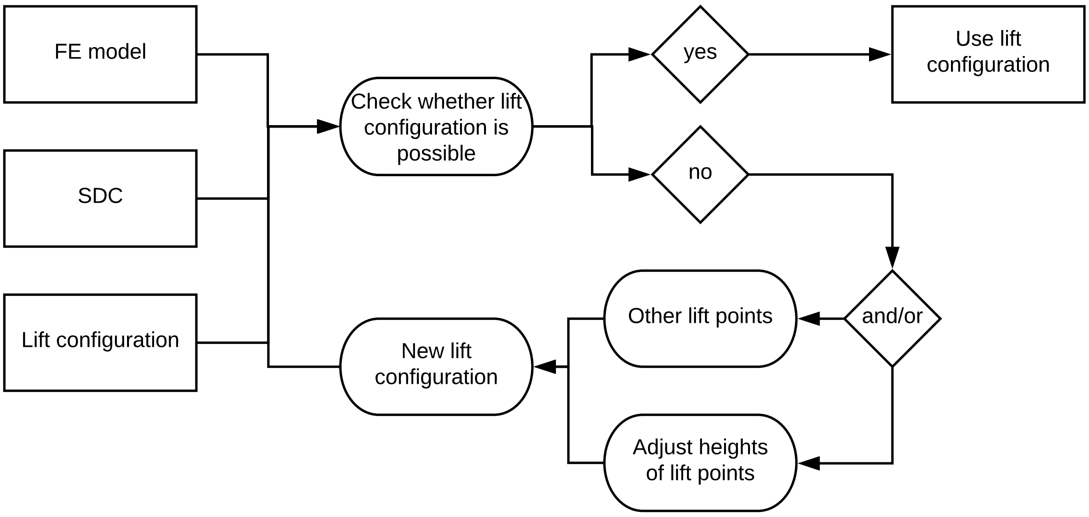


Figure 1.5: Current approach of designing a lift configuration for a single-lift operation

A new approach for designing a lift configuration is proposed, which is summarized by fig. 1.6. The main advantage of this new approach is, that instead of just proposing a feasible lift configuration, in this approach an optimal lift configuration is proposed. The finite element model and the SDC project are still used as input for designing a lift configuration. However, in this new approach the heavy lift engineer does not propose one individual lift configuration, but instead, the heavy lift engineer identifies all possible lift locations. Together with the FE model and the SDC project, the possible lift locations are then used in an optimization process.

In the optimization process, the goal is to minimize the amount of reinforcements and the amount of lift points. As there are an unlimited amount of possible lift configurations, it is impossible to try them all. An optimization strategy needs to be found, which can find an optimized lift configuration in a systematic way. For designing this optimization strategy, the characteristics of the optimization problem will be defined in section 1.3.3, possible problem solving strategies will be given in section 1.3.4 and section 1.3.5 explains the final chosen approach.

Resulting from this optimization process, one or more improved lift configurations are presented. Those configurations should be possible, but to be sure, will be checked for feasibility. If multiple lift configurations are provided, the lift configuration which fits best to the client's wishes is used. This can be based on costs, complexity, redundancy and safety margins on the utilizations of individual members.

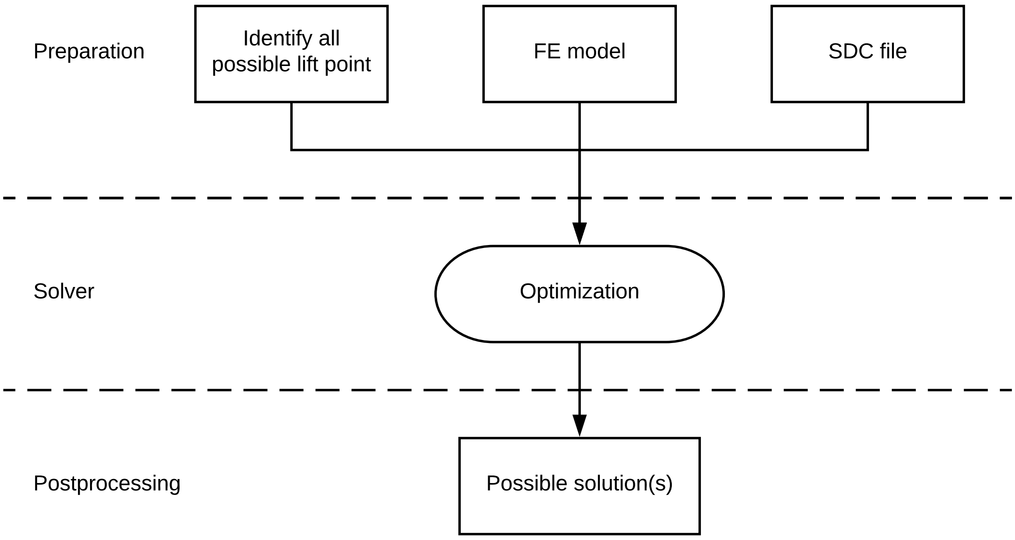


Figure 1.6: Proposed approach of designing a lift configuration for a single-lift operation

1.3.3. Characteristics of the problem

First of all, the problem is not restricted to a single topside, but could potentially embrace the removal of around 470 topsides [22] in the North Sea alone. This means that if someone wishes to develop a valuable method for optimizing the lift configuration of a topside during a single lift operation, this method should be easily applicable to new projects.

Finding an optimal lift configuration, as mentioned in section 1.3.1, is actually a minimization problem. The objective is to minimize both the amount of lift points used, and the amount of reinforcements done on the topsides. In this thesis an approach is chosen to reduce this minimization problem to a single-objective problem, by using a single cost function.

In section 1.3.2 it was defined, that all possible lift points are identified before the optimization process. This means that the first variable, whether a lift location is used or not, is an integer variable. The values for each lift points are binary: one for a used lift point or zero for an unused lift point. The other variable, the relative height of each lift point, is a continuous variable. Together, this forms a *mixed-integer* problem.

Removing or adding a lift point changes the entire force distribution within the topsides. Therefore, a change in the location of a lift point has a non-linear influence on the amount of reinforcements needed, and hence the minimization problem is *non-linear*.

As there is an unlimited number of possible combinations of lift configurations, not all lift configurations can be evaluated. By varying the height of a lift configuration which uses the same lift points, local minima can be found. However, it is not certain if a local minimum is also the global minimum. Moreover, since different combinations of lift points, as well as different amount of lift points can be used, it is quite complex to evaluate all possible combinations of lift points. This, in combination with the non-linearity of the problem, makes it a *non-convex* problem [7].

As explained in section 1.3.2, the lifting beams have a maximum lifting capacity, and thus the reaction forces have to be lower than this maximum. Furthermore in section 1.3.1 the minimum (two) and maximum (four) amount of lift points per side were stated. The optimal lift configuration must satisfy these constraints, and hence the minimization problem is *constrained*.

In summary, this minimization problem is:

- Mixed-integer
- Non-linear
- Non-convex
- Constrained

1.3.4. Possible problem solving strategies

There are a few possible problem solving strategies for a mixed-integer minimization problem. A frequently used approach for a minimization problem, is a nonlinear programming solver, such as the *fmincon* function from the Matlab optimization toolbox [26]. Fmincon is a derivative-based local minimum finder, which uses the second derivative of the object function. The advantage of using the fmincon algorithm, is the relatively short period it needs for finding an optimum. The disadvantage of this algorithm is that it needs a continuous objective function, which can be differentiated twice. This is not straightforward, as in the objective function the stresses and strains need to be calculated, for which an implementation of the finite element method (FEM) is needed. This could be solved by making a surrogate model [30]. In a surrogate model the original problem is mimicked by fitting a function through several data sets, in order to make a continuous function. However, it is expected that this approach is not easy to apply as a general method which can be used easily for different topsides, as a new surrogate model has to be made for every topside. Furthermore fmincon is still a local minimum searcher and not a global one.

The next option is to use a pattern search algorithm. The pattern search begins with an initial estimate provided by the user, which will be evaluated by the fitness function. The next step is to create new points based on the initial estimate, but with a higher and lower value for one variable at a time. These new points are then evaluated, and the point with the lowest fitness value becomes the new estimate. This means that new solutions are only found in the neighborhood of the old solution, and that for every step a lot of new combinations have to be tried out before a new step can be made. An advantage of the pattern search algorithm is that is does not need a continuous objective function. A disadvantage is that pattern search finds a local minimum.

A different possibility would be to use a genetic algorithm (GA). The GA is based on Darwin's theory of survival of the fittest. An initial population is generated randomly. This population is rated using the fitness function which has to be minimized. The best individuals are then marked as parents, which will be used to generate a new population. This new generation is produced by either making random changes to a single parent, or by combining a pair of parents into a new individual. This randomness increases the probability of finding a global minimum. An advantage of the GA, is that the next guesses are chosen at random by the algorithm, instead of basing on the derivatives of the fitness function, which implies that the fitness function does not have to be continuous. Another advantage of the GA is its robustness: it is easy applicable to new topsides. The GA is also capable of working with integer numbers. A disadvantage of the genetic algorithm is the relative high computational cost [6]. Another disadvantage of the GA is that is does not guarantee finding a global minimum.

Another possible algorithm for a minimization problem is a particle swarm optimization (PSO). Like the GA it does not need a continuous fitness function. Multiple starting points are generated which each have an initial velocity. The initial points are evaluated based on the fitness function. Then for each point a new position is created by combining the last position, speeds, its best location in the past and the best location of all particles. Subsequently, a new velocity is calculated based on this step. It also has the advantage that in can run in parallel processing. PSO is a global minimum searcher, but it is never certain if a global minimum is found. Although it is possible to encode a mixed-integer implementation of the PSO [3], the standard implementation in Matlab is not capable of handling mixed-integer problems [26].

The different strategies are summarized in table 1.1.

Table 1.1: Summary of the advantages and disadvantages of different problem solving strategies

Problem solving strategy	Advantages	Disadvantages
Non-linear programming solver	Fast	Continuous fitness function
		Needs extra intermediate steps
Pattern search algorithm	Non-continuous fitness function	No global minimum guaranteed
		Small search space
Genetic algorithm	Robustness	Computational cost
	Non-continuous fitness function	No global minimum guaranteed
Particle swarm optimization	Non-continuous fitness function	No global minimum guaranteed

1.3.5. Chosen problem solving strategy

The GA has been selected as optimization algorithm because of its proven capability of solving mixed-integer problems [26]. The implementation of a mixed-integer genetic algorithm in Matlab is tested to 20 test problems [11], and this implementation of a GA outperforms a PSO algorithm. [18] already concluded that a major advantage of a GA is that it will work on any combination of variable type, and showed this by applying it to a mixed-integer problem as well.

The capability of the GA to solve the warehouse-allocation problem has been showed [23]. The warehouse-allocation shows similarities to the problem of finding an optimal lift configuration, as the integer decision to open a warehouse can be compared to the decision if a lift point is used. The size of a flow from a warehouse is a continuous variable, and can be compared to the setting of the height of a lift point.

Furthermore [2, 27] showed the application of a GA to two other problems. [2] proved the possibility of optimizing structural support locations of a simple beam using a hybrid genetic algorithm. [27] demonstrated the capability of the GA to solve topology optimization problems. Topology optimization problems are somewhat similar to this problem as they also deal with maximum stresses in trusses, except that the variables are of a different type.

Another important reason for selecting GA as optimization algorithm, is the possibility of programming a fitness or cost function, which is not continuous. The GA generates a random solution, and calculates a value for that solution, instead of finding a solution based on the derivative of a function. This makes it possible to include the FEM procedure in the fitness function, resulting in the possibility to include stresses and buckling checks in the fitness value of a solution, and also makes sure the GA can solve non-linear problems.

The genetic algorithm is capable of working with constraints [26], formulated as inequality constraints, as shown in eq. (1.1). In this equation $\boldsymbol{x}$ is the vector containing the variables to be optimized. Furthermore it is possible to write a non-linear inequality constraint function, which makes it possible to include other types of inequality constraints. In this inequality constraint function, additionally a FEM procedure can be implemented, which allows the incorporation of forces or stresses calculated by the FEM procedure to be included in the inequality constraints.

$$\boldsymbol{A} \cdot \boldsymbol{x} \leqslant \boldsymbol{b} \tag{1.1}$$

1.4. Research question and objectives

To solve the problem mentioned above, the research question of this thesis is:

"Can the lift configuration of a topsides be optimized using a genetic algorithm?"

To answer this question, the following research objectives have been formulated:

- To choose the amount of lift points, the locations of lift points and the amount of force applied on each lift point in a systematic way.
- To implement a structural model, in order to assess how forces and stresses are distributed in a topside.

- To determine how many reinforcements are needed to make sure the topside stays intact during a single lift-operation, and where these reinforcements are needed.
- To quantify the cost of different reinforcements to be able to make a substantiated decision between reinforcing the structure and manufacturing more lift points.
- To provide a cost-effective lift configuration.
- The proposed method should be conveniently applicable to different topsides, as there are many removal projects in the future.

1.5. Thesis outline

In the next chapter the optimization process is explained, which includes fundamental theory on the FEM, the model optimization parameters, the formulation of the cost function, constraints of the TLS and theory and information on the implementation of the GA. Chapter 3 explains the actual model of a topsides. An analysis of the cost function is performed in chapter 4. Chapter 5 shows the results of the optimization of a single-lift project. Finally, chapter 6 consists of the conclusion and the recommendations.

2
Optimization process

2.1. Introduction

In this chapter the complete optimization process is presented. Section 2.2 explains the theory and implementation of the FEM. Section 2.3 gives the optimization parameters. Section 2.4 describes the formulation of the cost function. Following these sections, section 2.5 clarifies the constraints of the system. Finally, section 2.6 gives the theory and implementation of the GA.

Before starting with the optimization process, it is convenient to introduce the notation of a lift configuration used within this thesis. The vector **x**, is a possible lift configuration. The mathematical formulation of this vector is shown in eq. (2.1). n_{LP} is the amount of lift points, and a lift configuration contains two pieces of information per lift point (as described in section 1.3.3). The first piece of information is an integer variable (denoted by a_n), which describes whether a lift point is used (value of a_n is one) or not (value of a_n is zero). The second piece of information is a continuous variable (denoted by b_n), which contains the relative height of each lift point.

$$\begin{aligned} &\mathbf{x} = [a_1 \quad a_2 \quad \ldots \quad a_{n_{LP}} \quad b_1 \quad b_2 \quad \ldots \quad b_{n_{LP}}] \\ &\text{where:} \\ &\quad a_n \in \{0,1\} \end{aligned} \tag{2.1}$$

2.2. Implementation of the finite element method

In section 1.3.5 it was defined that a genetic algorithm will be used for solving the minimization problem. Although a full explanation of the genetic algorithm will be given in section 2.6, it is important to already note that for complicated problems, in earlier research [11] it was showed that the amount of function evaluations can range from 10,000 up to 300,000 on average. Hence a complicated cost or fitness function has a great impact on the speed of the algorithm. As the idea is to include the finite element procedure in the cost function, the time needed to run this procedure should be as small as possible.

The amount of degrees of freedom (DOF)'s in a FE model of a topsides can range from several tens of thousands to over 300,000[1]. To significantly reduce the computational time needed for every FE procedure, FE model order reduction is used [31, 32].

This section is divided in four parts. First, section 2.2.1 explains the principle of the FEM. Next, section 2.2.2 provides an explanation on how the superposition principle is used. Furthermore, section 2.2.3 gives the theory behind FEM order reduction, and defines how this is used in the optimization of a single-lift configuration. Finally, section 2.2.4 summaries how these procedures are combined and this functions in practice.

2.2.1. Principle of the finite element method

The finite element method is a technique which makes it possible to model the structural behavior of a topsides. Using this technique, forces and stresses within a structure, as well as reaction forces, are determined.

[1] This is concluded after checking several FE model of topsides available within Allseas

This is done by dividing the structure, in this case a topsides, into discrete parts or elements [34]. The displacements at the boundaries of this element can be determined by solving a system of equations. In practice, this means that a topsides is described by two matrices. The first matrix is the mass matrix, which defines the mass distribution in a topsides. The second matrix is a stiffness matrix, which characterizes the rigidity of a structure. In other words, the stiffness defines the relationship between forces and displacements.

A static system (without damping) is described by eq. (2.2), where $\boldsymbol{K}$ is the stiffness matrix of the system, $\boldsymbol{u}$ are the displacements of the system and $\boldsymbol{F}$ is the vector of forces applied to the system. In this case, the loads are caused by gravity, hence the force can be rewritten to a multiplication of the mass matrix of the system ($\boldsymbol{M}$) and the gravitational acceleration vector ($\boldsymbol{g}$).

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{F} = \mathbf{M} \cdot \mathbf{g} \tag{2.2}$$

In eq. (2.2) no constraints are introduced to the system yet, which means that rigid body motions are not prevented. Therefore, the stiffness matrix is singular [25]. This can be solved by imposing boundary constraints to the system. For a single-lift operation of a topsides, the boundary locations are located at the lift points. For now, it is assumed that these lift points are located at the legs, as shown in fig. 2.1. In each lift point, the displacement in vertical (z-) direction is fixed. The complete structure also needs three restraints in the horizontal plane (x- and y-direction), to prevent rigid body motions in both horizontal directions (x- and y-direction), and rotational motions around the vertical (z-) axis. For simplicity, these boundary conditions are not shown.

At each junction of beams, a node is present which describes the displacements of that junction. Each node has 6 DOF ($\boldsymbol{u}$), 3 translations and 3 rotations. These DOF's are divided into two groups: the specified DOF's $\mathbf{u}_s$ (from which the displacement is specified) and the free DOF's $\mathbf{u}_f$ (which are free to move). The stiffness matrix is also divided between specified and free DOFs (eq. (2.3)). $\mathbf{F}_f$ are the loads corresponding to the free DOFs (in this case defined by multiplying the reduced mass matrix with the gravitational acceleration) and $\mathbf{F}_s$ is the vector containing the support reactions at the lift points (and thus at the boundary conditions).

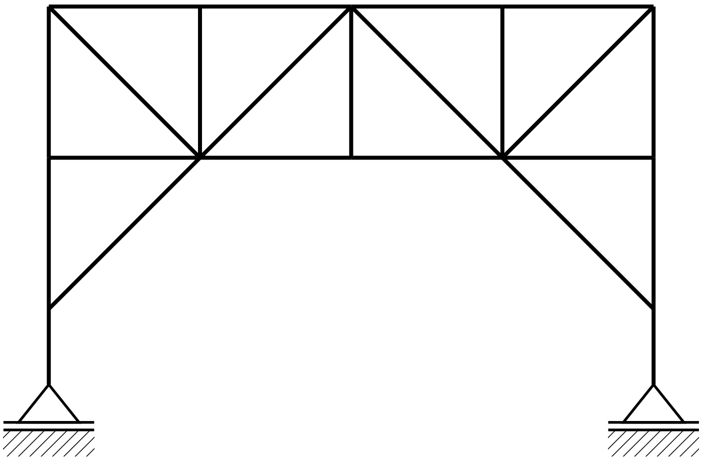


Figure 2.1: Boundary conditions on a simplified model of a topsides

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fs} \\ \mathbf{K}_{sf} & \mathbf{K}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_s \end{bmatrix} = \begin{bmatrix} \mathbf{F}_f \\ \mathbf{F}_s \end{bmatrix} \tag{2.3}$$

The displacements of the free DOF's can be determined from the upper part of eq. (2.3). Rewriting this part of the equation leads to eq. (2.4). The support reactions at the specified DOF's, thus the lift points, can be determined by using the lower part of eq. (2.3), which leads to eq. (2.5).

$$\mathbf{u}_f = -\mathbf{K}_{ff}^{-1}(\mathbf{K}_{fs} \cdot \mathbf{u}_s - \mathbf{F}_f) \tag{2.4}$$

$$\mathbf{F}_s = \mathbf{K}_{sf} \cdot \mathbf{u}_f + \mathbf{K}_{ss} \cdot \mathbf{u}_s \tag{2.5}$$

The displacements at the lift points are described, and together with the calculation of the displacements of the free DOF's, all displacements are known. This is used to derive the shear forces, axial forces and moments in each element. The DOF's are converted to the displacements of an element ($\boldsymbol{u}^e$). This is multiplied in eq. (2.6) with the element stiffness matrix ($\boldsymbol{k}^e$) in order to retrieve the vector with element forces $\boldsymbol{F}^e$ (which contains the axial forces, shear forces and moments in that element).

$$\boldsymbol{F^e} = \mathbf{k^e} \cdot \mathbf{u^e} \tag{2.6}$$

The relationship between displacements and strains can be described by eq. (2.7), where $\boldsymbol{\epsilon}$ are the strains of an element and $\boldsymbol{B}$ is the strain-displacement matrix. To get the stresses of the elements, the strains need to be multiplied with the stress tensor $\boldsymbol{C}$ (eq. (2.8)), defined by Hooke's Law [5]. This stress tensor differs per element type, but includes the material properties.

$$\boldsymbol{\epsilon} = \mathbf{B} \cdot \mathbf{u} \tag{2.7}$$

$$\boldsymbol{\sigma} = \mathbf{C} \cdot \boldsymbol{\epsilon} \tag{2.8}$$

2.2.2. Superposition principle

As only linear elastic behavior is considered [31], the superposition principle is valid. The superposition principle states that the response of a structure to the application of a system of forces and displacements is identical to the summation of the same structure, to the separate application of every force and displacement to the system. This principle is shown in fig. 2.2 for a single beam, where the total reaction of the system (right side of fig. 2.2), is equal to the sum of the reaction to the gravity force (on the left side of fig. 2.2), and the reaction to an imposed displacement.

The same principle is valid for the FE model of a complete topsides. In eq. (2.4) it is defined that the displacement of the unconstrained DOF's is a summation of two parts: a reaction to the gravity force on the system ($-\mathbf{K}_{ff}^{-1} \cdot \mathbf{F}_f$) and a reaction to the displacement of the specified nodes ($-\mathbf{K}_{ff}^{-1} \cdot \mathbf{K}_{fs} \cdot \mathbf{u}_s$). This is used for the calculation of the FE procedure of the topsides. First all possible lift points are fixed and a gravity load is added. Then, the reaction of the system to this arrangement is determined, which includes the reaction forces and forces and stress in each element. Subsequently, the reaction to a lift configuration is calculated. In each lift configuration, as defined in section 1.3.3, some of the possible lift points are used, and the heights of those lift points are specified. The other lift points are free. Equation (2.4) is used to calculate the displacement of the free lift points, and eq. (2.3) is used to calculate the reaction forces. The total reaction force for the free lift points sums up to zero, which works out with having no boundary conditions at those lift points. The stresses and internal forces are also summed.

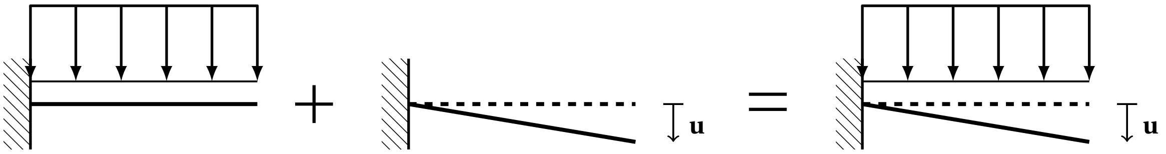


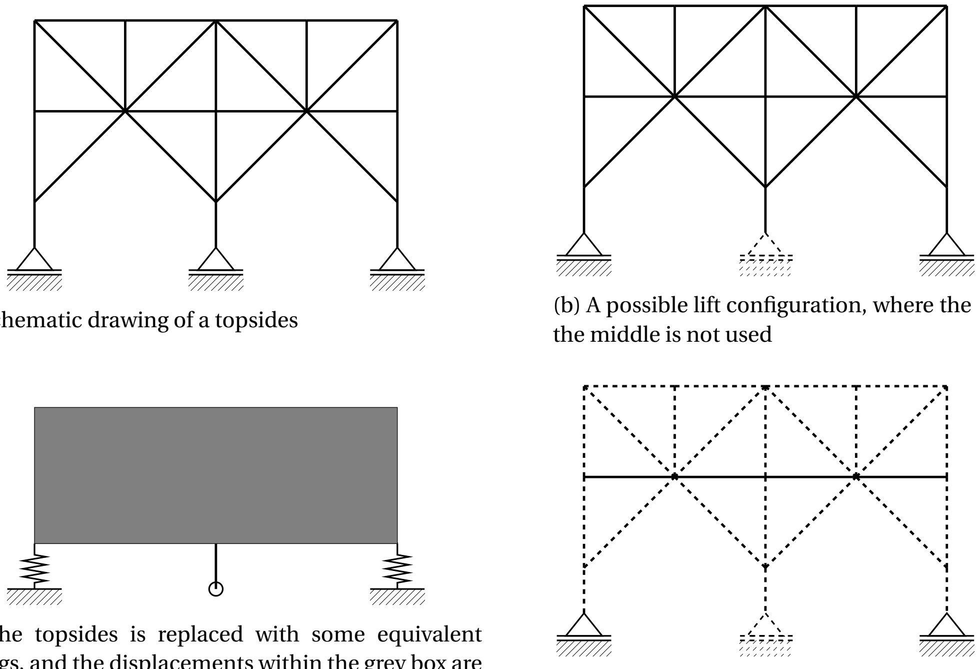
Figure 2.2: The superposition principle on a single beam

2.2.3. Finite element model order reduction

As a complete FE model can have over 100,000 DOF's, each calculation will become time consuming. Finite element model order reduction is used to significantly reduce the computational time needed for every calculation [32]. Finite element model order reduction, also known as static reduction, is a technique to reduce the amount of DOF's.

Using the finite element method, the displacements of all nodes can be determined according to eq. (2.3). However, for evaluating a lift configuration of a topsides, the displacements within the topsides are actually not of any interest. The resulting force and stress distribution is the essential information of a lift configuration.

An example of a topsides is shown in fig. 2.3a. For a topsides, the only DOF's of interest, are the displacements in vertical direction at the lift points. Furthermore, as explained in section 2.2.1, 3 extra DOF's are needed to constrain rigid body motions. In fig. 2.3b, one of the lift points is unconstrained. To be able to calculate the displacement of this free lift point, the displacement of the other nodes (not located at the lift point) are unnecessary. The complete topsides is replaced by some equivalent springs, which only relate the displacements of the lift points. To achieve this, the mass and stiffness matrix are reduced to the nodes of the lift points. The result is shown in fig. 2.3c.



(a) Schematic drawing of a topsides

(b) A possible lift configuration, where the lift point in the middle is not used

(c) The topsides is replaced with some equivalent springs, and the displacements within the grey box are disregarded

(d) Part of the topsides of which forces and stresses are evaluated, using the displacements of the lift points

Figure 2.3: Schematic drawing of finite element model order reduction

Degrees of freedoms which are not of any interest, are removed by rewriting eq. (2.2). First the stiffness matrix and the DOF's are divided between retained ($\mathbf{u}_r$) and condensed DOF's ($\mathbf{u}_c$ - eq. (2.9), [5]). The lower part of this equation can be solved for $\mathbf{u}_c$ (eq. (2.10)). The multiplication of $-\mathbf{K}_{cc}^{-1}\mathbf{K}_{cr}$ in this equation is defined as the static transformation matrix $\boldsymbol{G_{ot}}$ (eq. (2.11)). This matrix maps the retained DOF's u_r to the condensed DOF's u_c.

$$\begin{bmatrix} \mathbf{K}_{rr} & \mathbf{K}_{rc} \\ \mathbf{K}_{cr} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_r \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{F}_r \\ \mathbf{F}_c \end{bmatrix} \tag{2.9}$$

$$\mathbf{u}_c = -\mathbf{K}_{cc}^{-1}(\mathbf{K}_{cr}\mathbf{u}_r - \mathbf{F}_c) \tag{2.10}$$

$$\mathbf{G}_{ot} = -\mathbf{K}_{cc}^{-1}\mathbf{K}_{cr} \tag{2.11}$$

The expression for $\mathbf{u}_c$ can then be substituted in the upper part of eq. (2.9), this leads to eq. (2.12), which is rewritten to split the forces from the DOFs (eq. (2.13)). The reduced stiffness matrix $\mathbf{K}_{red}$ is then obtained (eq. (2.14)).

$$\mathbf{K}_{rr} * \mathbf{u}_r - \mathbf{K}_{rc}\mathbf{K}_{cc}^{-1}(\mathbf{K}_{cr}\mathbf{u}_r + \mathbf{F}_c) = \mathbf{F}_r \tag{2.12}$$

$$(\mathbf{K}_{rr} - \mathbf{K}_{rc}\mathbf{K}_{cc}^{-1}\mathbf{K}_{cr})\mathbf{u}_r = \mathbf{F}_r - \mathbf{K}_{rc}\mathbf{K}_{cc}^{-1}\mathbf{F}_c \tag{2.13}$$

$$\mathbf{K}_{red} = \mathbf{K}_{rr} - \mathbf{K}_{rc}\mathbf{K}_{cc}^{-1}\mathbf{K}_{cr} \tag{2.14}$$

The mass matrix also needs to be mapped to a reduced mass matrix. This can be done by summing the retained part of the mass matrix $\mathbf{M}_{rr}$ with the transformed parts of the remaining mass matrix. The static transformation matrix $\boldsymbol{G_{ot}}$, which relates the retained DOFs with the condensed DOFs can be used for this transformation. The influence which the condensed part of the mass matrix $\mathbf{M}_{cc}$ has on the retained DOF's, can be obtained by pre- and post multiplying $\mathbf{M}_{cc}$ with $\boldsymbol{G_{ot}}$. The off-diagonal terms $\mathbf{M}_{rc}$ and $\mathbf{M}_{cr}$ can be transferred to the retained part by multiplying them once with the static transformation matrix. The complete description of the reduced mass matrix $\mathbf{M}_{red}$ can be found in eq. (2.15) [31].

$$\mathbf{M}_{red} = \mathbf{M}_{rr} + \mathbf{M}_{cr}\mathbf{G}_{ot} + \mathbf{G}_{ot}^{T}\mathbf{M}_{rc} + \mathbf{G}_{ot}^{T}\mathbf{M}_{cc}\mathbf{G}_{ot} \tag{2.15}$$

Using eq. (2.3), eq. (2.14) and eq. (2.15), the reaction forces and displacements of the lift points are known. As a full model has many members, a selection will be made to only check the stresses of important members. In fig. 2.3d four members in the middle are selected, while for a full model members which are important for the structural integrity will be chosen. The stresses of these members are determined by multiplying the displacement of the free nodes with the static transformation matrix. This is then used together with eqs. (2.6) to (2.8) to obtain the stresses in the selected elements.

2.2.4. Summary of the complete finite element procedure

In fig. 2.4 a summary of the finite element procedure is given. First, possible lift points and critical members are identified. This is used to run a gravity case, where the static stresses and reaction forces are calculated. The mass and stiffness matrix are then reduced, in order to only retain the DOF's at the lift points. A possible lift configuration is provided, and by using this lift configuration the displacements of the lift points are calculated. These displacements are used for two matters: to calculate the reaction forces at the lift points and to calculate the total reaction forces. The total reaction forces can be calculated by using the superposition principle to sum up the reaction forces at the lift points and the reaction forces of the gravity case. Next, the relation between the lift points and the critical members is defined, which is used together with the displacements of the lift points to calculated the forces and stresses in the critical members. Finally, this is summed with the results from the gravity case to obtain the total forces and stresses in the critical elements.

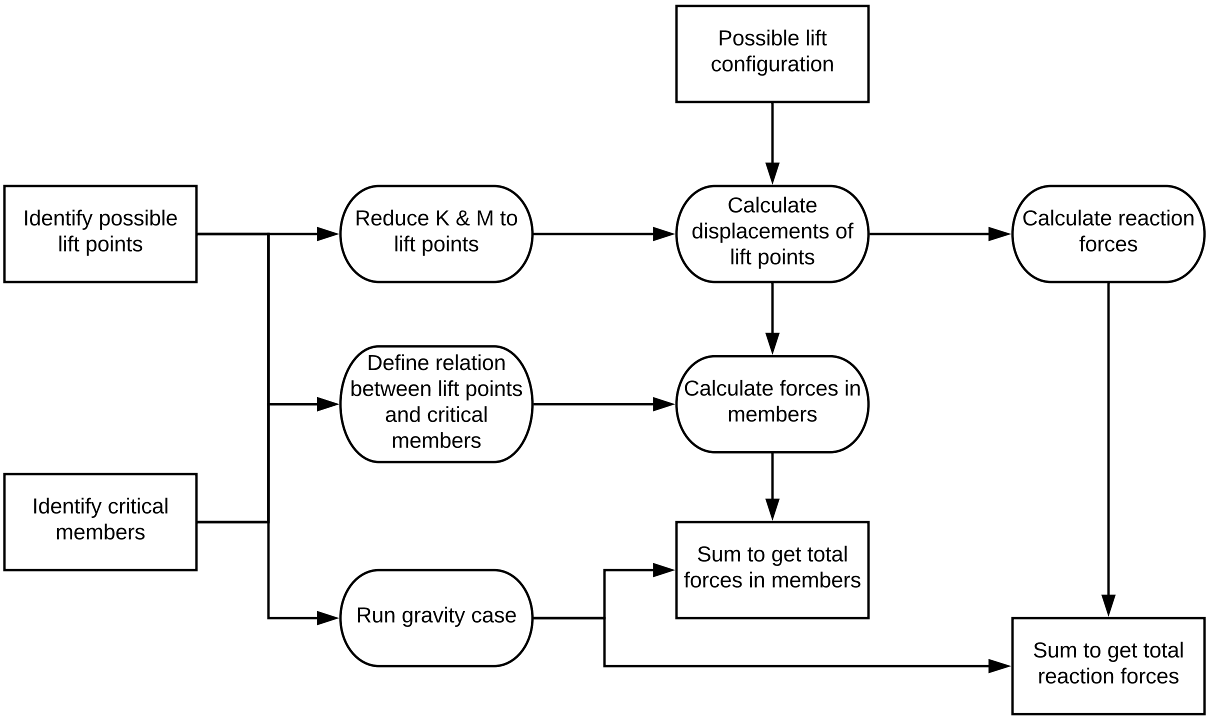

Possible lift configuration
Identify possible lift points
Reduce K & M to lift points
Calculate displacements of lift points
Calculate reaction forces
Define relation between lift points and critical members
Calculate forces in members
Identify critical members
Sum to get total forces in members
Run gravity case
Sum to get total reaction forces

Figure 2.4: Summary of the finite element procedure

2.3. Model optimization parameters

After the forces, stresses and moments in each member are calculated as described in section 2.2, the next step is to verify if a member will survive these conditions. Members will be checked for different failure mechanisms, including axial failure, bending failure, shear failure and combined failure [33]. The members of the structure are divided in two groups: tubular members and other members. Tubular member will be checked in section 2.3.1, while other members will be checked in section 2.3.2.

2.3.1. Tubular members

Tubular member will be checked according to the ISO 19902 [21] and all formulas in this section are obtained from this standard unless stated otherwise. Members subjected to axial tension will be evaluated against their yield strength, while members in compression will not only be checked for yielding, but also for buckling. Finally members under shear stress will be checked for shear failure. The total maximum utilization is equal to the maximum of the utilizations explained below.

Members under tension

The utilization of tubular members under axial tension is calculated using eq. (2.16). σ_t is the axial tensile stress as calculated in eq. (2.8), f_y is the yield strength of the material of the member and $\gamma_{R,t}$ is the partial resistance factor for axial tensile strength and is equal to 1.05. The utilization of a member (u_j) should be lower than 1.0 to make sure the member does not fail.

$$u_{j,axial} = \frac{\sigma_t}{f_y/\gamma_{R,t}} \tag{2.16}$$

Members under compression

For tubular members under axial compression first the buckling strength has to be determined. In eq. (2.17) the representative elastic local buckling strength (f_{xe}) of a member is calculated. C_x is the elastic critical buckling coefficient for which a reduced value of 0.3 [21] is used. E is the Young's modulus of the material and D and t are respectively the outside diameter and wall thickness of the member.

$$f_{xe} = \frac{2C_x E t}{D} \tag{2.17}$$

The elastic local buckling strength is then used in eq. (2.18) to calculate the representative local buckling strength (f_{yc}), which combines the resistance against buckling and the yield strength into a combined strength. This representative local buckling strength is used in eq. (2.19) to calculate the representative axial compressive strength f_c. Equation (2.19) uses the column slenderness parameter λ as defined in eq. (2.20). In this equation K is the effective length factor as defined in ISO 19902 and has a value between 0.7 and 1. L is the length of a member and r_g is the radius of gyration.

$$f_{yc} = \begin{cases} f_y & \text{if } \frac{f_y}{f_{xe}} \leqslant 0.170, \\ (1.047 - 0.274\frac{f_y}{f_{xe}})f_y & \text{if } 0.170 < \frac{f_y}{f_{xe}} \end{cases} \tag{2.18}$$

$$f_c = \begin{cases} (1.0 - 0.278\lambda^2)f_{yc} & \text{if } \lambda \leqslant 1.34, \\ \frac{0.9}{\lambda^2} f_{yc} & \text{if } 1.34 < \lambda \end{cases} \tag{2.19}$$

$$\lambda = \frac{KL}{\pi r_g}\sqrt{\frac{f_{yc}}{E}} \tag{2.20}$$

The final step is to calculate the utilization of a member (u_j), which is done in eq. (2.21). In this equation σ_c is the axial compressive stress (as calculated in eq. (2.8)) and f_c is the representative strength of the member (as defined in eq. (2.19)). $\gamma_{R,c}$ is the partial resistance factor for axial compressive strength and is equal to 1.18 [21]. The utilization of a member should be lower than 1.0.

$$u_{j,axial} = \frac{\sigma_c}{f_c/\gamma_{R,c}} \tag{2.21}$$

Bending failure

The bending strength of a tubular member can be obtained from eq. (2.22). In this equation Z_p is the plastic section modulus and Z_e is the elastic section modulus. The bending strength can be used to calculated the utilization of a member under beam shear according to eq. (2.23). M is the maximum bending moment and $\gamma_{R,b}$ is the partial resistance factor for bending strength and is equal to 1.05 [21].

$$f_b = \begin{cases} \left(\frac{Z_p}{Z_e}\right) f_y & \text{if } \frac{f_y D}{Et} \leqslant 0.0517, \\ \left[1.13 - 2.58\left(\frac{f_y D}{Et}\right)\right]\left(\frac{Z_p}{Z_e}\right) f_y & \text{if } 0.0517 < \frac{f_y D}{Et} \leqslant 0.1034 \\ \left[0.94 - 0.76\left(\frac{f_y D}{Et}\right)\right]\left(\frac{Z_p}{Z_e}\right) f_y & \text{if } 0.1034 < \frac{f_y D}{Et} \end{cases} \tag{2.22}$$

$$u_{\boldsymbol{j},bending} = \frac{M\gamma_{R,b}}{f_b Z_e} \tag{2.23}$$

Members under combined loading

If a member is under tension, the combined loading is equal to the axial loading (eq. (2.16)). If a member is under compression, the combined utilization is the highest value of the two equations in eq. (2.24). The first equation is just a summation of the axial and compressive utilization. The second equation takes buckling into account, and the second part of the equation is combination of the compressive and bending stress. In this equation, C_m is the moment reduction factor in each direction and f_e is the Euler buckling strength.

$$u_{\boldsymbol{j},combined} = \begin{cases} u_{\boldsymbol{j},axial} + u_{\boldsymbol{j},bending} \\ u_{\boldsymbol{j},axial} + \frac{\gamma_{R,b}}{f_b}\left[\left(\frac{C_{m,y}\,\sigma_{b,y}}{1-\sigma_c/f_{e,y}}\right)^2 + \left(\frac{C_{m,z}\,\sigma_{b,z}}{1-\sigma_c/f_{e,z}}\right)^2\right]^{0.5} \end{cases} \tag{2.24}$$

Shear failure

The utilization of a member under beam shear can be calculated according to eq. (2.25) and should be lower than 1.0. V is the beam shear and $\gamma_{R,v}$ is partial resistance factor for shear strength (and is equal to 1.05 [21]). f_v is the representative shear strength, which is equal to $\frac{f_y}{3}$.

$$u_{\boldsymbol{j},shear} = \frac{2V\gamma_{R,v}}{f_v A} \tag{2.25}$$

2.3.2. Other members

Non-tubular member are checked according to the AISC 360-10 [1] and all formulas described in this section are obtained from this standard unless stated otherwise. All utilizations should be lower than 1.0. The total maximum utilization is equal to the maximum of the utilizations calculated in this section.

Members under axial forces

Members under tension are checked according to eq. (2.26), while members under axial compression are checked according to eq. (2.29). In eq. (2.26) $\gamma_{R2,t}$ is the tension resistance factor, which is equal to 0.9 [1].

$$u_{\boldsymbol{j},axial} = \frac{\sigma_t}{f_y \gamma_{R2,t}} \tag{2.26}$$

Members under axial compression have to be checked for buckling. The elastic buckling stress can be calculated as stated in eq. (2.27). In this equation $\frac{KL}{r_g}$ is the member slenderness. r_g is the radius of gyration and K is the effective length factor. Using the elastic buckling stress the critical stress F_{cr} can be calculated in eq. (2.28). The critical stress can be compared to the stress in a member to calculate the utilization of that member (eq. (2.29). $\gamma_{R2,c}$ is the partial resistance factor for axial compressive strength and is equal to 0.9 for other members.

$$F_e = \frac{\pi^2 E}{\left(\frac{KL}{r_g}\right)^2} \tag{2.27}$$

$$F_{cr} = \begin{cases} \left(0.658^{\frac{f_y}{F_e}}\right) f_y & \text{if } \frac{f_y}{F_e} \leqslant 2.25, \\ 0.877 F_e & \text{if } 2.25 < \frac{f_y}{F_e} \end{cases} \tag{2.28}$$

$$u_{\boldsymbol{j},axial} = \frac{\sigma_c}{F_{cr}\gamma_{R2,c}} \tag{2.29}$$

Members under shear forces

Members which are under a shear force, have to be checked for shear failure. In eq. (2.30) the nominal shear strength is calculated. For an I-beam this has to be done for both the flange and the web area. C_v is the shear coefficient. In eq. (2.31) the shear utilization is calculated. $\gamma_{R2,v}$ is the partial resistance factor for shear strength, which is equal to 0.9 and F_v is the shear force in a member.

$$V_n = 0.6 f_y A_f C_v \tag{2.30}$$

$$u_{\boldsymbol{j},shear} = \frac{F_v}{V_n \gamma_{R2,v}} \tag{2.31}$$

Members under a bending moment

The utilization for bending can be calculated according to eq. (2.32). This has to be done for bending in both the major and the minor axis. M_n is the nominal flexural strength of member and $\gamma_{R2,b}$ is the partial resistance factor for bending strength and is equal to 0.9 for other members.

$$u_{\boldsymbol{j},bending} = \frac{M}{M_n \gamma_{R2,b}} \tag{2.32}$$

Members under combined loading

Members subject to both axial loading and a bending moment have a combined utilization as calculated in eq. (2.33). This is a combination of the utilization for axial loading, the utilization for bending in major axis and the utilization for bending in minor axis.

$$u_{\boldsymbol{j},combined} = \begin{cases} u_{\boldsymbol{j},axial} + \frac{8}{9}(u_{\boldsymbol{j},bending,major} + u_{\boldsymbol{j},bending,minor}) & \text{if } 0.2 \leqslant u_{\boldsymbol{j},axial}, \\ 0.5 * u_{\boldsymbol{j},axial} + u_{\boldsymbol{j},bending,major} + u_{\boldsymbol{j},bending,minor} & \text{if } u_{\boldsymbol{j},axial} < 0.2 \end{cases} \tag{2.33}$$

2.4. Formulation of the cost function

In order to make a decision if a lift configuration is a good lift configuration, an automatic way of judging a lift configuration is introduced. This is done by defining a cost function. The cost function analyses a solution and comes with a price to achieve this solution. This results in the possibility to compare multiple solutions. The cost which have to be made to be able to lift a topside can roughly be divided in two categories: the cost for constructing a lift point (further explained in section 2.4.1) and the cost for reinforcing overloaded beams (further explained in section 2.4.2).

2.4.1. Costs for lift points

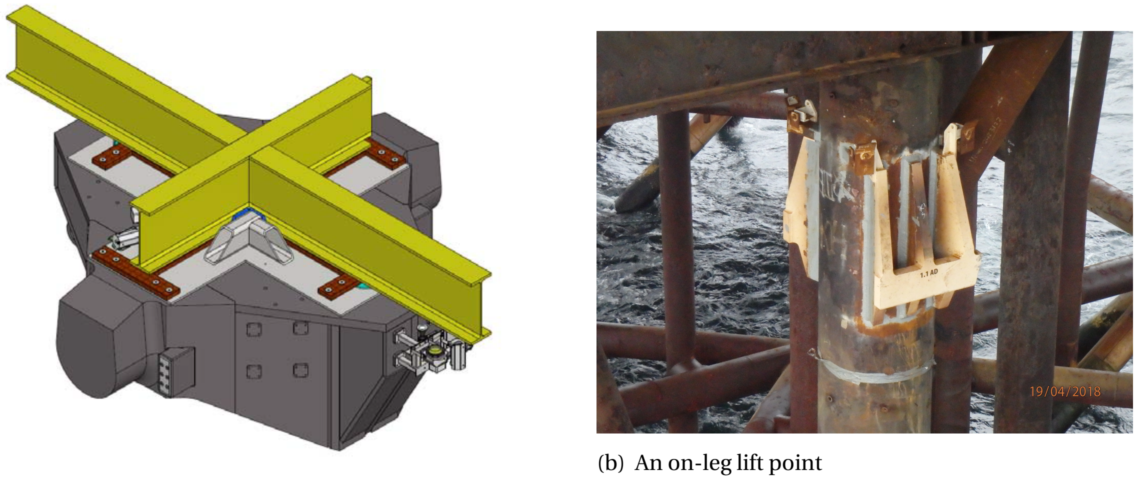
Lift points can be divided in two categories. The first category is an under beam lift point, of which an impression can be found in fig. 2.5a. For this kind of lift point are in principle little reinforcements needed. The production of the supports can be done on-shore, which makes this kind of lift point relatively cheap.

The second category is a lift point placed on a leg, as shown in fig. 2.5b. For this kind of lift point a lifting frame has to be created on-shore, which has to be attached to the legs offshore. Because the majority of the work has to be done on shore, the price of this type of lift point is relatively cheap, but more expensive than the under beam lift point.

In table 2.1 the investment per individual lift point is stated. As a leg-type lift point has been used more often, the cost of this type of lift point is well known. The cost for other types of lift point is more uncertain. The total cost for lift points is obtained by summing the investments for each lift point.

Table 2.1: Investment per type of lift point

Type of lift point	Cost [€]
Under beam	100,000
Leg	250,000



(a) An under-beam lift point

(b) An on-leg lift point

Figure 2.5: Different type of lift points

2.4.2. Cost for different reinforcements

Utilizations of members are calculated in accordance with section 2.3.1 for tubular members, and in accordance with section 2.3.2 for other type of members, using the forces and stresses in each element as described in section 2.2.1. As these utilizations are dependent on a lift configuration, the costs are also dependent on a lift configuration. In the likely case some utilizations are higher than 1.0, reinforcements have to be done. According to these utilizations, three failure modes can be identified: buckling, yielding and shear deformation.

In order to make an estimation of the cost for reinforcement, a general approach for reinforcements is made. Based on this approach a volume of steel can be determined, which can be multiplied with the cost per kilo offshore steel (c_{steel}). According to knowledge within the company this c_{steel} should roughly be equal to 25 euro. This cost includes all the work which has to be done offshore as well as the cost to get the material onto the topside.

The reinforcements are estimated for tubular members, for I-beams and for rectangular tubes, as these types are the most used ones. If multiple reinforcements have to be made on a member, only the most expensive one is done, as it is assumed that this reinforcement solves all utilizations. In addition, because repairs offshore are expensive, a lower limit of the costs per reinforcement has been set to €5,000. This is in order to reduce the amount of small reinforcements. When a lift configuration is chosen, a detailed assessment has to be made for the reinforcements. The cost for each reinforcement is denoted by $C_{rf,j}$, where j is the number of the element.

In general, for the length of the beam which needs reinforcement, a distinction is made between local and global reinforcements. In section 1.3.2 it was explained that SDC verifier finds the complete buckling length of an element, which can be longer than the element itself. For buckling failures, thus members under axial compression, the full buckling length will be taken as length, while for other types of failure the length of the element is taken. For a combined failure the buckling length is taken if the axial utilization is at least 25% of the bending utilization.

Tubular members

Members under axial forces
The utilization for tubular members under axial forces has been determined in eq. (2.16). To lower the stresses in a member, the cross-sectional area is increased. An example how this is done can be found in fig. 2.6. The cost for this is calculated according to eq. (2.34), where ρ_{steel} is the density of steel.

$$C_{rf,\boldsymbol{j}}(\mathbf{x}) = c_{steel} \cdot (u_{\boldsymbol{j},failuretype}(\mathbf{x}) - 1) \cdot A \cdot L \cdot \rho_{steel} \tag{2.34}$$

Members under a bending moment
The utilization for bending has been determined in eq. (2.23). To increase the bending resistance (eq. (2.22)), the plastic section modulus will be increased. This is done by adding stiffeners, as shown in fig. 2.6. A conservative way to calculate the total size of the stiffeners which should be added, is by multiplying the area with the utilization, as shown in eq. (2.34). The axial utilization is replaced by the bending utilization in this case. In practice this will increase the plastic section modulus with an even bigger factor.

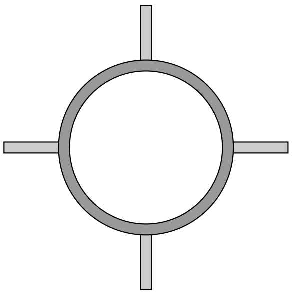


Figure 2.6: Tubular members reinforced with stiffeners

Members under a combined loading
The utilization for members under combined loading has been determined in eq. (2.24). As both axial and bending failures are resolved in the same way, this is also used for the combined loading case. A conservative way of calculating the volume of added steel, is to use the combined utilization as a multiplication factor (eq. (2.34)), as in practice adding stiffeners decreases both the axial utilization as the bending utilization.

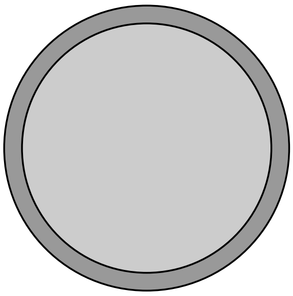


Figure 2.7: Tubular members reinforced with concrete

Members under shear forces
If tubular members fail under shear force, the shear forces need to be redistributed. This is done by filling the tube with concrete, as shown in fig. 2.7. The cost is then calculated according to eq. (2.35), where $c_{concrete}$ and $\rho_{concrete}$ are respectively the cost factor per kilo and the density of concrete.

$$C_{rf,\boldsymbol{j}}(\mathbf{x}) = c_{concrete} \cdot A_{inner} \cdot L \cdot \rho_{concrete} \tag{2.35}$$

I-beams

Members under axial forces
The utilization under axial forces has been calculated in eq. (2.26) and eq. (2.29). If an utilization needs to be reduced, this is achieved by increasing the cross-sectional area of an I-beam, as shown in fig. 2.9 and fig. 2.10. The cost for this is shown in eq. (2.34).

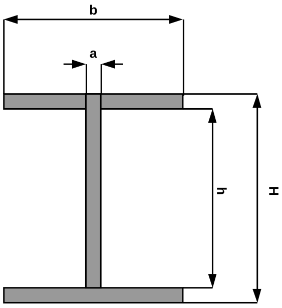


Figure 2.8: Dimensions of an I-beam

Members under a bending moment in the major axis
In eq. (2.32) the utilization for bending has been calculated. If this utilization exceeds 1.0, this is reduced by increasing the plastic section modulus in major axis, as shown in fig. 2.9. The symbols for the dimensions of an I-beam are shown in fig. 2.8, and T_f is the flange thickness. The plastic section modulus is calculated according to eq. (2.36), which is used to calculate the factor (λ_{flange}), with which the flange thickness needs to be thickened. λ_{flange} is calculated according to eq. (2.37). This factor is used to calculate the cost according to eq. (2.38).

$$Z_x = b \cdot T_f \cdot (H - T_f) + 0.25 \cdot a \cdot h^2 \tag{2.36}$$

$$\lambda_{flange} = \frac{u_{\boldsymbol{j},bending,major}(\mathbf{x}) \cdot Z_x - 0.25a \cdot h^2}{b \cdot T_f(H - T_f)} \tag{2.37}$$

$$C_{rf,\boldsymbol{j}}(\mathbf{x}) = c_{steel} \cdot (\lambda_{flange} - 1) \cdot b \cdot T_f \cdot L \cdot \rho_{steel} \tag{2.38}$$

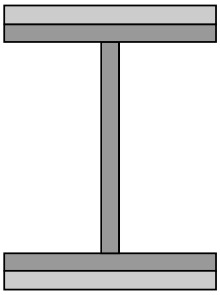


Figure 2.9: Reinforcements made on the flanges of an I-beam

Members under a bending moment in the minor axis
For an over-utilization in the minor axis, the plastic section modulus in the minor axis (eq. (2.39)) needs to be increased. This is done by reinforcing the web of an I-beam, as shown in fig. 2.10. The thicknesses of the plates are calculated according to eq. (2.40), which leads to the cost shown in eq. (2.41).

$$Z_y = b^2 \frac{T_f}{2} + 0.25 \cdot a^2 \cdot h \tag{2.39}$$

$$a2 = \frac{(u_{\boldsymbol{j},bending,minor}(\mathbf{x}) - 1) \cdot Z_y}{0.25 \cdot b \cdot h} \tag{2.40}$$

$$C_{rf,\boldsymbol{j}}(\mathbf{x}) = c_{steel} \cdot a2 \cdot h \cdot L \cdot \rho_{steel} \tag{2.41}$$

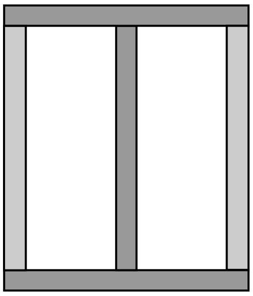


Figure 2.10: Reinforcements made on the web of an I-beam

Members under a combined loading
The utilization for combined loading can be found in eq. (2.33). For solving an utilization which is higher than 1.0, the reinforcements shown in fig. 2.9 will be done. Because reinforcing the flange will both lower the utilization in bending and the utilization in axial direction, a new combined utilization has to be calculated. This will be done by increasing the flange thickness by 1% until the total utilization becomes lower than 1.0. The cost is then calculated according to eq. (2.42), where $\lambda_{combined}$ is the total increase in flange thickness.

$$C_{rf,\boldsymbol{j}}(\mathbf{x}) = c_{steel} \cdot (\lambda_{combined} - 1) \cdot A_{flange} \cdot L \cdot \rho_{steel} \tag{2.42}$$

Members under shear forces
The utilization for shear has been calculated in eq. (2.31). If a flange failure happens, a reinforcement is made as shown in fig. 2.9 and the cost is calculated according to eq. (2.44). If a web failure happens, it is solved according to fig. 2.10 and eq. (2.44).

$$C_{rf,\boldsymbol{j}}(\mathbf{x}) = c_{steel} \cdot A_{flange} \cdot L \cdot \rho_{steel} \tag{2.43}$$

$$C_{rf,\boldsymbol{j}}(\mathbf{x}) = c_{steel} \cdot A_{web} \cdot L \cdot \rho_{steel} \tag{2.44}$$

Rectangular tubes

Members under axial forces
For rectangular tubes the utilization under axial forces has been calculated in eq. (2.26) and eq. (2.29). If an utilization is higher than 1.0, this is solved by increasing the area as shown in fig. 2.12. The cost for this is calculated in eq. (2.34).

Members under a bending moment
In eq. (2.32) the utilization for bending has been calculated. If this utilization is higher than 1.0, for rectangular tubes this will be solved by increasing the plastic section modulus, as shown in fig. 2.13. This reinforcement will only be done in the direction where the bending moment is too high. The plastic section modulus is calculated according to eq. (2.45). The symbols for the dimensions of the rectangular tube are shown in fig. 2.11. Using the moment of inertia, λ_{tb} is calculated (eq. (2.46)), where λ_{tb} is the factor by which the top and bottom plates are increased. This is subsequently used to estimate the cost of extra steel as shown in eq. (2.47). The same procedure is also followed for solving bending in the other axis.

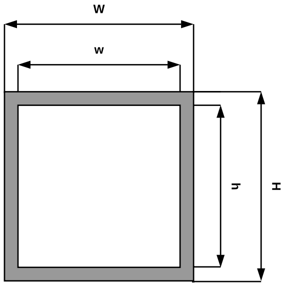


Figure 2.11: Dimensions of a rectangular tube

$$Z_x = 0.25 \cdot \left(H^2 \cdot W - h^2 \cdot w\right) \tag{2.45}$$

$$\lambda_{tb} = \frac{1}{H}\sqrt{(u_{\mathbf{j},bending}(\mathbf{x}) \cdot Z_x + h^2 \cdot w)\frac{4}{W}} \tag{2.46}$$

$$C_{rf,\mathbf{j}}(\mathbf{x}) = c_{steel} * \lambda_{tb} \cdot 2 \cdot H \cdot W \cdot L \cdot \rho_{steel} \tag{2.47}$$

Members under a combined loading
The utilization for combined loading is calculated in accordance to eq. (2.33). Combined problems will be solved as shown in fig. 2.13. If the height of the plates on the top and bottom are increased with a factor $\lambda_{r,comb}$, the utilization for axial forces and the utilization for bending in both axes will be lowered. This makes it necessary to calculate a new combined utilization. To solve this problem the plate thickness will be increased in steps of 1%. The final costs are calculated according to eq. (2.48).

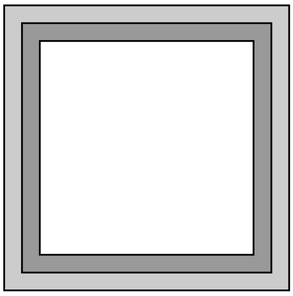


Figure 2.12: Reinforcements made on all edges of a rectangular tube

$$C_{rf,\mathbf{j}}(\mathbf{x}) = c_{steel} \cdot (\lambda_{r,comb} - 1) \cdot A_{top+bottom} \cdot L \cdot \rho_{steel} \tag{2.48}$$

Members under shear forces
Shear utilization had been calculated in eq. (2.31). If this value is higher than 1.0, this can be decreased by increasing the cross-sectional area as shown in fig. 2.12. The cost for the reinforcement is estimated by eq. (2.34).

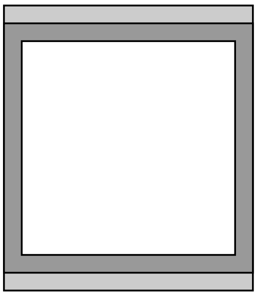


Figure 2.13: Reinforcements made on two edges to increase bending resistance

2.4.3. Combined costs

In eq. (2.49) the complete cost function can be found, where in the first part the costs of all lift points are summed. In the second part of eq. (2.49) the costs of reinforcements is summed over all n_{elem} elements, where d_j makes sure that only the costs for reinforcements of members that actually need a reinforcement will be taken into account.

$$C(\mathbf{x}) = \sum_{i=1}^{n_{LP}} C_{LP,\boldsymbol{i}} \cdot \mathbf{x}[i] + \sum_{j=1}^{n_{elem}} C_{rf,\boldsymbol{j}}(\mathbf{x}) \cdot d_{\boldsymbol{j}}$$

where: (2.49)

$$d_{\boldsymbol{j}} = \begin{cases} 0 \text{ if } 1 < u_{\boldsymbol{j}}(\mathbf{x}), \\ 1 \text{ if } u_{\boldsymbol{j}}(\mathbf{x}) \geqslant 1 \end{cases}$$

2.5. Geometrical and capacity constraints

The amount of possible lift configurations is limited by the capacities of the TLS. In section 1.2 the geometry of the TLS was already explained. To guarantee a stable lift configuration and to prevent toppling of a platform, a minimum of two lift points per side is used. As there are only eight pairs of lifting beams, a maximum of four lift points are used per side. Another geometrical constraint is caused by the size of the lifting beams. It is possible to identify multiple lift points close to each other. However, as enough space is needed to fit forklift units of lifting beams next to each other, a minimum center to center distance of 6.1 meter is needed between two lift points. If two possible lift points are located closer to each other, only one of them can be used at the same time.

In section 1.3.1 it was defined that for a given lift configuration, the relative height of each lift point can be adjusted. In section 1.2 an explanation was given on how this is done in practice. The limitations of the TLS can be found in table 2.2, which shows that the maximum relative height difference between different lift points, is 0.208 meter. The maximum difference in relative height has to be lower than this.

Table 2.2: The achievable height limits of the TLS

Description	Transport
Minimum height [m]	0.383
Maximum height [m]	0.591
Achievable stroke [m]	0.208

The maximum loads on the TLS are limited by two mechanisms: the maximum force and the maximum moment in the lifting beams. The maximum force in z-direction on a pair of lifting beams is 72 MN. As the maximum bending moment in a lifting beam is also limiting, the maximum force is dependent on the outreach of the lifting beams. Because the moment in the lifting beams is dependent on the outreach of the beams, the capacity of the beams decreases if the distance of a lift point with the center line of the vessel becomes smaller. The maximum capacity of the TLS is shown in fig. 2.14.

2.6. The Genetic Algorithm

A GA is a search algorithm, introduced by [20], which can be used for optimization problems. Genetic algorithms are based on Darwin's (and Spencer's) principle of survival of the fittest, and form an attempt to implement this principle in a smart way, in an algorithm. According to the principle of survival of the fittest, favorable individuals have a higher chance of passing along their genes to a new generation. The intention is to find the best feasible individual. A clear description on how to implement a binary genetic algorithm has been given by [16] and a summary of the procedure can be found in fig. 2.15. Furthermore in appendix A an explanation on a self-made genetic algorithm is found, and in appendix B the implementation of this algorithm in Matlab is shown.

In a genetic algorithm a population of possible solutions (the first generation) is generated. A population in a genetic algorithm is encoded in such a way that the important characteristics of an individual can be judged, based on (predefined) criteria. This is incorporated in a fitness or *cost function*, which calculates a value for every individual. In this way the entire population can be evaluated in a fair way. A new population (the next generation) will be generated based on three major operators: *selection*, *crossover* and *mutation*. Individuals in the old generation are also called parents, while individuals in the new generation are called children.

In this case, an individual, is a possible lift configuration (**x**). The mathematical formulation of an individual is stated in eq. (2.50), and the maximum relative height difference described in table 2.2 are added. For

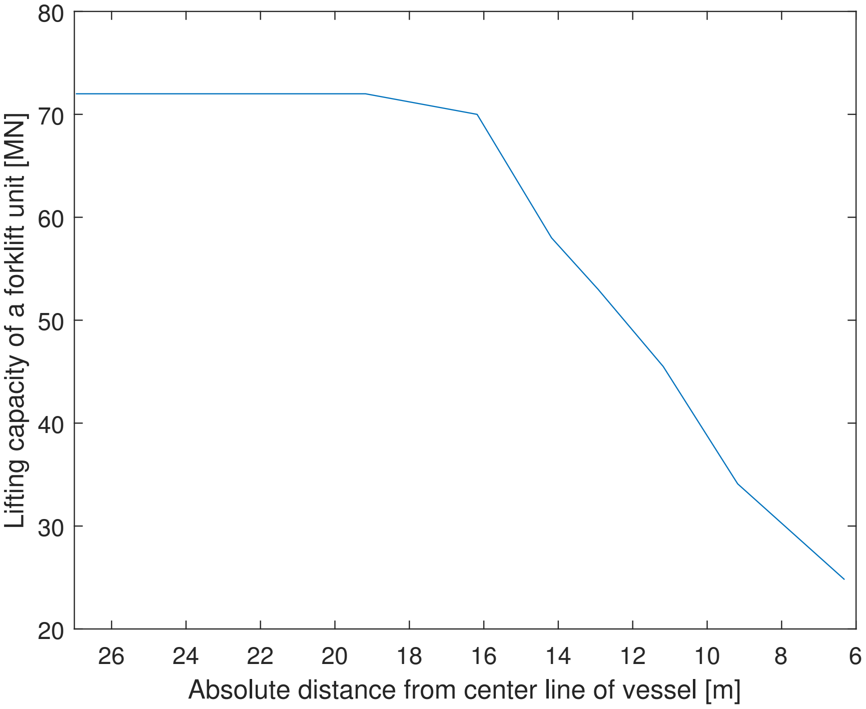


Figure 2.14: Lifting capacity of a forklift unit compared to the distance of the center line of the vessel [data retrieved from internal communications within Allseas]

each individual a FE procedure will be done, which is explained in further detail in section 2.2. The reaction forces in the lift points are determined using eq. (2.5). Furthermore the amount of reinforcements needed for each lift configuration is determined according to section 2.3.

$$\begin{aligned} & x = [a_1 \quad a_2 \quad \dots \quad a_{n_{LP}} \quad b_1 \quad b_2 \quad \dots \quad b_{n_{LP}}] \\ & \text{subject to:} \\ & \qquad a_n \in \{0,1\} \\ & \qquad -0.104 \le b_n \le 0.104 \end{aligned} \tag{2.50}$$

2.6.1. Minimization problem

The goal of this optimization problem is to minimize the investments needed for a lift configuration. In eq. (2.49) a cost function was defined, and the objective is to minimize this cost function, as stated in eq. (2.51). In eqs. (2.53) to (2.54) the maximum amount of lift points per side is set to four, corresponding with section 2.5. The minimum amount of lift points is set to two in eqs. (2.55) to (2.56). The reaction force in each lift point (calculated according to the procedure in eq. (2.5)), should be lower than the capacity of the TLS on that lift point (section 2.5), which is stated in eq. (2.57). Finally in eq. (2.58) a procedure is stated, which is used if two lift points (m and $m+1$) are located too close to each other to be used at the same time.

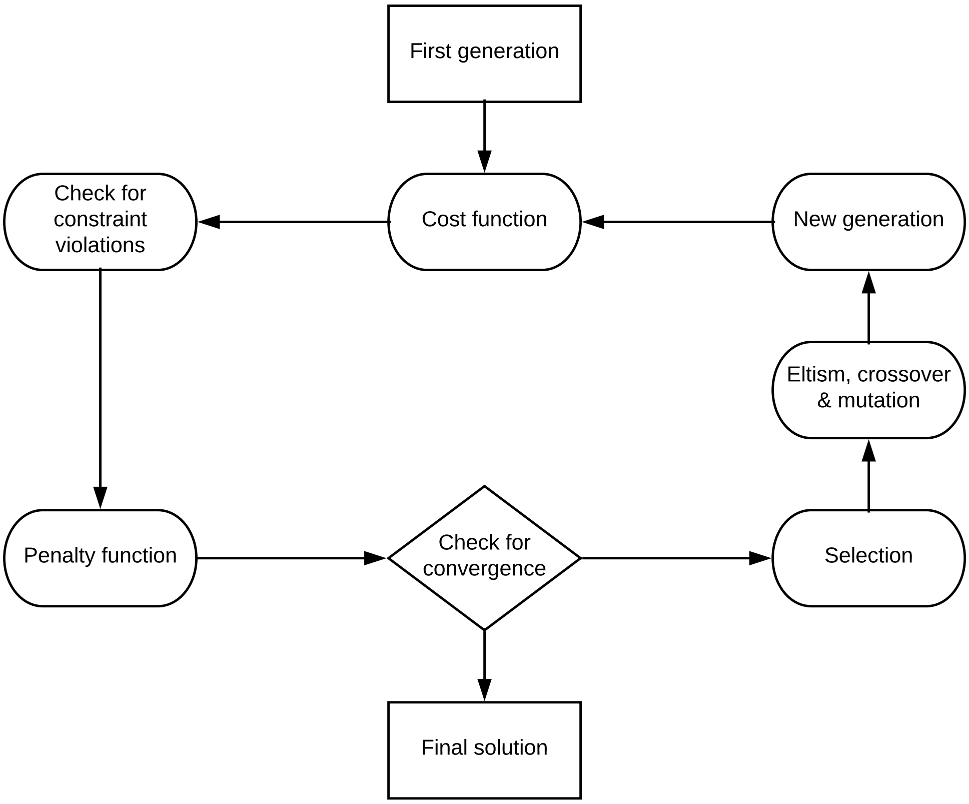


Figure 2.15: Summary of the procedure of a genetic algorithm

$$\min \ C(\mathbf{x}) = \sum_{i=1}^{n_{LP}} C_{LP,\boldsymbol{i}} \cdot \mathbf{x}[i] + \sum_{j=1}^{n_{elem}} C_{rf,\boldsymbol{j}}(\mathbf{x}) \cdot d_{\boldsymbol{j}} \tag{2.51}$$

where:

$$d_{\boldsymbol{j}} = \begin{cases} 0 \text{ if } 1 < u_{\boldsymbol{j}}(\mathbf{x}), \\ 1 \text{ if } u_{\boldsymbol{j}}(\mathbf{x}) \geqslant 1 \end{cases} \tag{2.52}$$

subject to:

$$\sum_{\boldsymbol{k}=1}^{\frac{1}{2}n_{LP}} \mathbf{x}[k] \leqslant 4, \tag{2.53}$$

$$\sum_{\boldsymbol{k}=\frac{1}{2}n_{LP}+1}^{n_{LP}} \mathbf{x}[k] \leqslant 4, \tag{2.54}$$

$$2 \leqslant \sum_{\boldsymbol{k}=1}^{\frac{1}{2}n_{LP}} \mathbf{x}[k], \tag{2.55}$$

$$2 \leqslant \sum_{\boldsymbol{k}=\frac{1}{2}n_{LP}+1}^{n_{LP}} \mathbf{x}[k], \tag{2.56}$$

$$F_{r,LP}(\mathbf{x}) \leqslant F_{TLS,cap,LP}, \tag{2.57}$$

$$\mathbf{x}[m] + \mathbf{x}[m+1] \leqslant 1 \tag{2.58}$$

In each generation, a population of lift configurations is present. A part of this population violates one or more of the constraints stated in eqs. (2.53) to (2.58).The individuals are divided in two groups: a group with feasible solutions and a group with unfeasible solutions. The group with feasible solutions keeps their value determined by the cost function, while for the group with unfeasible solutions, their values get replaced with a penalty value. This is achieved by a penalty function eq. (2.59). The penalty function starts with the value of the worst feasible individual of that generation (C_{worst}), to make sure a feasible solution is always preferred over an unfeasible solution [8]. Next, a penalty (P_n) is added for each constraint violation ($n_{constraints}$). The penalty becomes bigger, if a constraint violation becomes bigger. This means that between two unfeasible solutions, the one with the the smallest sum of penalties has the lowest value and is preferred.

$$C(\mathbf{x}) = C_{worst} + \sum_{n=1}^{n_{constraints}} P_n(\mathbf{x}) \tag{2.59}$$

2.6.2. Selection

Individuals in the old generation are called parents, and individuals in the new generations are called children. The selection procedure decides which individuals remain and which individuals are used for reproduction. According to [35], *elitism* can help in finding an optimal solution. Elitism means that the best individuals of the population are copied to the next generation, to make sure the algorithm (also) keeps searching in the neighborhood of the best solutions found so far. In this case the best 2 % of the population is copied to the next generation.

Different mechanisms can be used to select which individuals should be able to reproduce. According to [17] tournament selection is the most efficient method for selecting parents for a new generation. In this technique the whole population is added twice to a selection pool. All individuals are randomly paired with another individual. From these two individuals, the lift configuration with the lowest value is added to the mating pool, from which new individuals will be generated later. In this way individuals with a better fitness value have a higher probability to be used for reproduction.

In the mating pool two individuals are randomly paired. There is a $p_{crossover}$ probability that crossover (which is described in section 2.6.3) happens, otherwise mutation (which is described in section 2.6.4) is forced. The complete reproduction procedure is visualized in fig. 2.16.

2.6.3. Crossover

Just like reproduction in nature, a genetic algorithm uses crossover to produce two new individuals based on two old individuals. Crossover means that the new individuals get characteristics of old ones. In this implementation of the genetic algorithm, crossover is implemented as described by [9]. At first two random parents (${x_i}^{(1)}$ and ${x_i}^{(2)}$) are selected from the mating pool, where i represents the number of the current generation and the number in superscript represents the number of each individual. Next a random number β is generated according to the Laplace distribution. Finally two new individuals (children) (${x_{i+1}}^{(1)}$ and ${x_{i+1}}^{(2)}$) are created as described in eqs. (2.60) to (2.61), where β decides how close a child is to each parent.

In fig. 2.17 a crossover example is shown, where an individual exists of two continuous variables (b_1 & b_2). A pair of parents, already selected from the mating pool, is shown in red. New individuals, the children, will be created somewhere along the line between the two parents. The exact location is determined by the value of β, and in this figure possible locations are shown.

For integer variables (a_n), the numbers which come out of eqs. (2.60) to (2.61) are changed into integer values [11]. For integer variables which can only be zero or one, this means that there is a probability of 0.5 that the new variable becomes 0 or 1.

$$x_{i+1}^{(1)} = x_i^{(1)} + \beta|x_i^{(1)} - x_i^{(2)}| \tag{2.60}$$

$$x_{i+1}^{(2)} = x_i^{(2)} + \beta|x_i^{(2)} - x_i^{(1)}| \tag{2.61}$$

2.6.4. Mutation

Mutation is introduced to maintain diversity in a population. Because of the selection, elitism and crossover procedures described in this chapter, a population tends to create new offspring in the proximity of the best individuals of the last generation. However, it is possible that the current best individual is only a local optimum, and not the global optimum. Mutation adds a random element to the generation of a new population, and in this way tries to make sure a global optimum will be found.

Mutation is forced if crossover does not take place. Two complete individuals (which were together in the mating pool) will be mutated. Each individual is mutated separately, where the integer variables (a_n) are

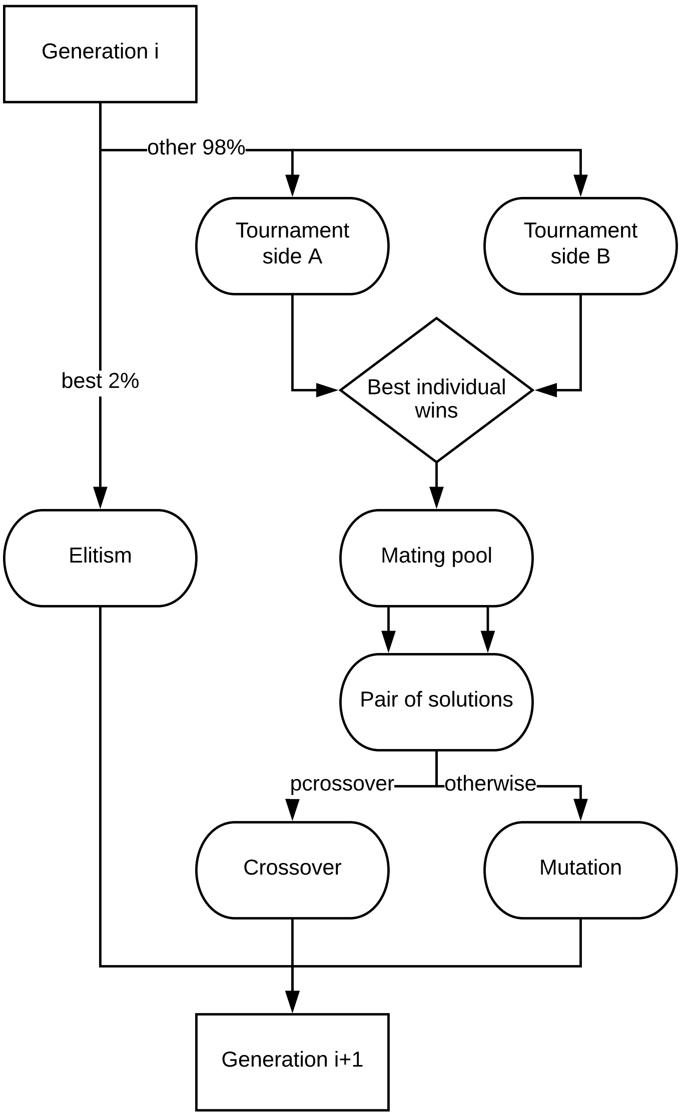


Figure 2.16: Reproduction procedure of the genetic algorithm

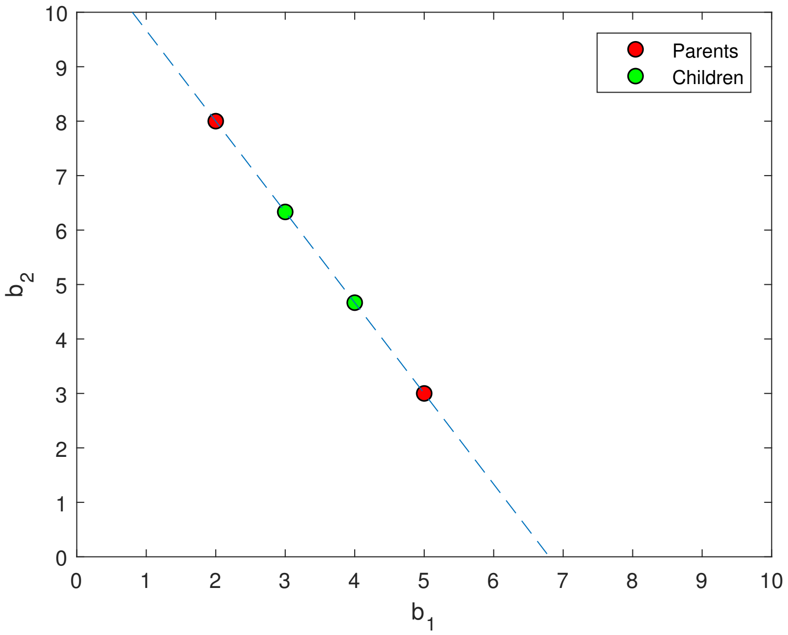


Figure 2.17: Example of the crossover procedure with 2 continuous variables

mutated according to the left side of fig. 2.18 [11], and the continuous variables (b_n) are mutated according to the right side of fig. 2.18 [10].

For integer values, just as in section 2.6.3, this means that they get a new random value of zero or one. The continuous variables of an individual are mutated according to eq. (2.62) (Deep and Thakur [10]). s is a random number according to the power distribution and r is a uniform random number between zero and one. i is the number of the generation (thus b_n^i is a continuous variable from the parent, a b_n^{i+1} from the child). l is the lower limit of each variable and u is the upper limit of each variable (in this case respectively the minimum and maximum relative height of each lift point).

$$b_n^{i+1} = \begin{cases} b_n^i - s(b_n^i - b_n^{i,l}) & \text{if } t < r, \\ b_n^i - s(b_n^{i,u} - b_n^i) & \text{if } t \geqslant r \end{cases}$$

where: (2.62)

$$t = \frac{b_n^i - b_n^{i,l}}{b_n^{i,u} - b_n^i}$$

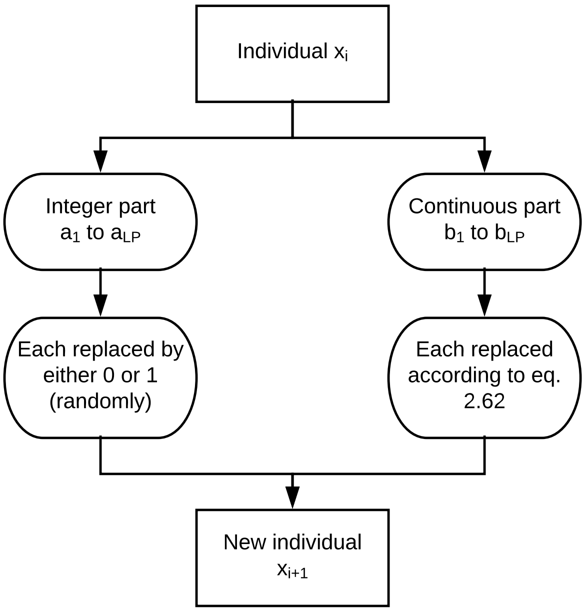

Individual x_i
Integer part a_1 to a_{LP}
Continuous part b_1 to b_{LP}
Each replaced by either 0 or 1 (randomly)
Each replaced according to eq. 2.62
New individual x_{i+1}

Figure 2.18: Reproduction procedure of the genetic algorithm

3

Case study on Gyda topsides

Section 3.1 introduces the topsides used for testing the new procedure. Section 3.1.1 identifies possible lift point locations and section 3.1.2 states which members are checked for reinforcements. Section 3.2 gives an explanation on how the boundary conditions are included. Next, section 3.3 gives the current lift configuration.

3.1. Topsides to be removed with a single-lift operation

A case study will be conducted on the Gyda topsides (fig. 3.1), a platform in the Southern part of the Norwegian sector of the North Sea. The important characteristics of the topside can be found in table 3.1. The dimensions are the outer dimensions, the width of the main support frame (MSF) is only 25 meters. The mass reported is the total mass that is to be removed, which has to be lifted by Pioneering Spirit.

Table 3.1: Characteristics of the Gyda topsides

Dimensions (l x w x h) [m]	93.8 x 56.0 x 56.5
Dry mass [ton]	17,806
CoG location [m]	[295.9,-0.1,60.6]

A FE model of the Gyda platform is available and can be found in fig. 3.2. The FE model has a total of 42,306 DOF's. The self-weight of the topsides ($L_{self-weight}$), is factored according to eq. (3.1), which leads to a factored load $L_{ULS,lift}$. As the assessment of the mass are not foolproof, the mass will be multiplied with a weight contingency factor (F_{weight})[13]. There is also some uncertainty in the location of the center of gravity (CoG), for which a CoG variance factor (F_{CoG}) is used [12]. The dynamic loads during the lift of a topside are replaced with a dynamic load factor (F_{dyn}). The structural integrity of the members of the model, which will be checked according to the procedure described in section 2.3, will be verified according to a ultimate limit state (ULS) condition A [15]. This means ULS A load factor (F_{ULSA}) will be used on the self weight. The values for these safety factors can be found in table 3.2, while a clear overview of all safety factors can be found in appendix C.

$$L_{ULS,lift} = F_{weight} \cdot F_{CoG} \cdot F_{dyn} \cdot F_{ULSA} \cdot L_{self-weight} \tag{3.1}$$

Table 3.2: Safety factors used in the model

Safety factor	symbol	value [-]
Weight contingency factor	F_{weight}	1.1
CoG variance factor	F_{CoG}	1.1
Dynamic load factor	F_{dyn}	1.15
ULS A factor	F_{ULSA}	1.2
SLS factor	F_{SLS}	1.0

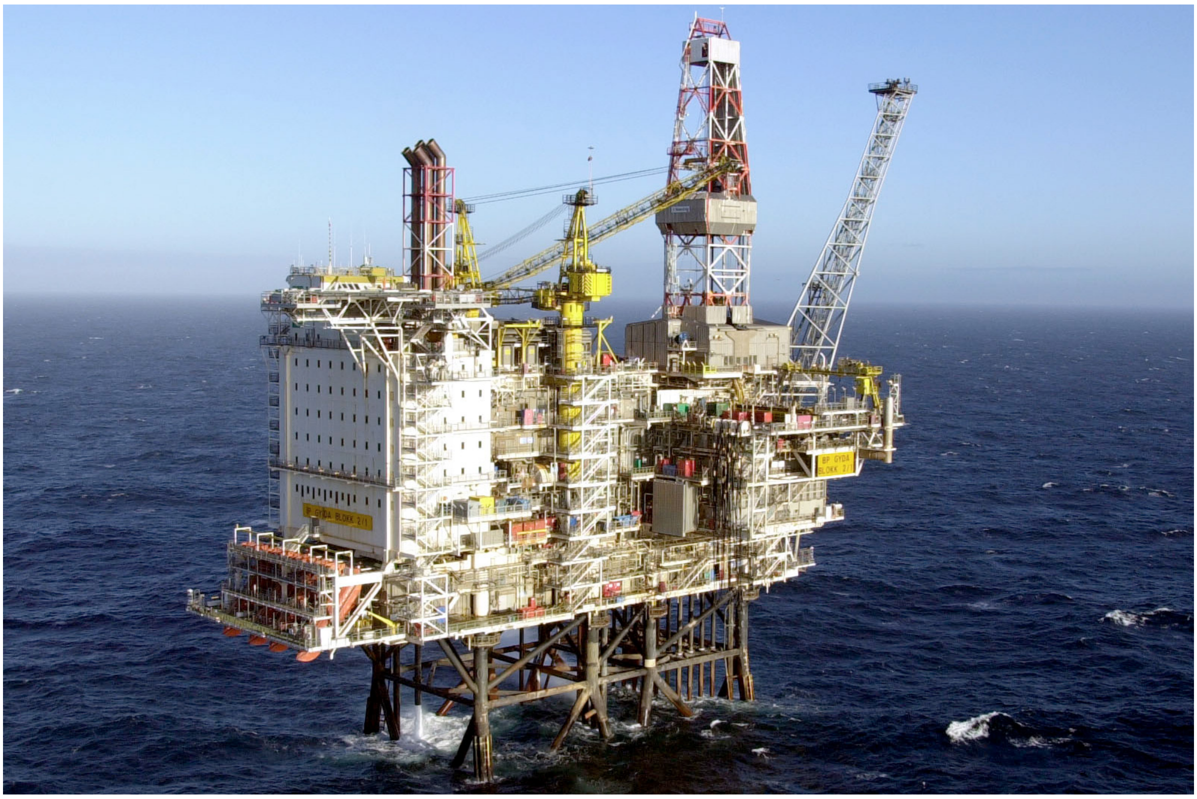


Figure 3.1: The Gyda topsides [retrieved from http://www.align.no/successful-gyda-project/]

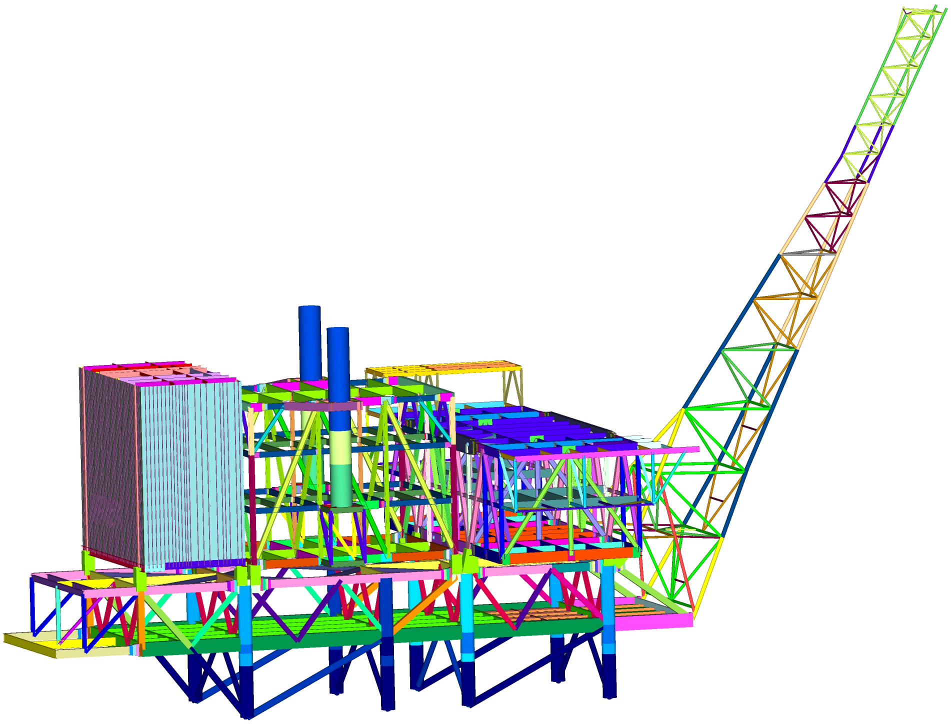


Figure 3.2: The Gyda topsides in FEMAP

3.1.1. Identifying possible lift points

The first step for a single-lift project, as proposed in fig. 1.6, is to identify possible lift location. For identifying lift points, the picture in fig. 3.1, information from the FE model in fig. 3.2 and information about the TLS (section 1.2) will be combined.

The first logical lift points, are the eight legs of the platform, as these have already been designed to pass on loads from the weight of the platform to the jacket. For identifying possible under beam lift points, it is important the TLS does actually fit at the location where someone wants to place a lift point. This greatly restricts the amount of possible lift locations. Furthermore, the static weight distribution has been studied. The reaction forces at the legs, when the topsides is still standing on the jacket, are shown in table 3.3. The numbering of the legs is done according to fig. 3.4.

Table 3.3: Reaction forces at legs during static condition

Lift point	A2	A3	A5	A6	B2	B3	B5	B6
Reaction force [MN]	39.1	3.8	24.2	25.7	38.9	4.0	25.0	26.1

Two of the legs in the middle, A3 & B3, have a small reaction force. The two other legs in the middle, A5 & B5, also have a load which is not too high compared to the capacity of the topsides lift system (fig. 2.14. It might be possible to replace these 4 on-leg lift points in the middle of the platform by two under-beam lift points, located in between the 2 legs on each side of the platform.

The two legs on the left side of fig. 3.1, A2 & B2, have the highest loading of all legs. This is mainly caused by the loading of the module in the middle of the platform, while also the load of the living quarters is, for the most part, transferred to these two legs. To be able to relieve the lift points on those two legs of a part of their loading, extra under-beam lift points close to these two are added. The preferred location would be the joints to the left of these legs, however, as a beam goes down to the legs from those joints, an under-beam lift point does not fit on that location. In fig. 3.2 (left side of the figure) it is visible that the dimensions of the members change at the next joint are of different sizes (see the difference between the yellow and green member, and the small members located in between them). Because of this change in member size, locating lift points at this joint seems impossible. As a compromise, two lift points are located between the two joints (A1 & B1, fig. 3.4).

The 2 intersections at main deck level between legs A1 & A2 and A3 & A4, would be two (or four if the B-side of the topsides is included as well) excellent locations for under-beam lift points. However, a study has been done in the FE model to check whether a pair of lifting beams fit above the diagonal member underneath those intersections, which resulted in rejecting these intersections as possible lift points.

The mass and stiffness matrices of the topsides are then reduced according to the procedure described in section 2.2.2. Twelve DOF's in the vertical (z-)direction are retained at the lift points, while 3 DOF's in horizontal direction are retained to prevent rigid body motions (section 2.2.1). The total size of the mass and stiffness matrix is reduced from two square matrices of order 42,306 to two square matrices of order 15, while keeping the information about the stiffness and mass of the whole topsides. In section 2.2.3 it was explained that using finite element model order reduction is used to significantly reduce the amount of time needed for every FEM calculation. If a sparse solver is used, (which is already much more effective than solving the complete matrix), the reduced computational effort would at least be of the order $\frac{42306}{15} = 2820$ [34].

3.1.2. Topsides members needed for structural integrity

In section 2.2.2 it was defined that only a part of the topsides is checked for reinforcements. This is done partly to reduce the amount of calculations, but also because not all members are important for the structural integrity of the topsides. In this case the MSF will be evaluated, as those beams are essential for the structural integrity of the total structure. Furthermore, these beams are of major importance to divide the loads through the structure. For other parts of the structure it might be less interesting if a beam yields, as the structure will not be reused. Information needed for the calculation of the utilizations (section 2.3) of the beams of the MSF will be exported from FEMAP [31]. In fig. 3.3 the remaining part of the FE model can be found. The relationship between the remaining DOF's (at the lift points) and the forces and stresses in the the remaining beams is found according to eq. (2.11) and eqs. (2.6) to (2.8).

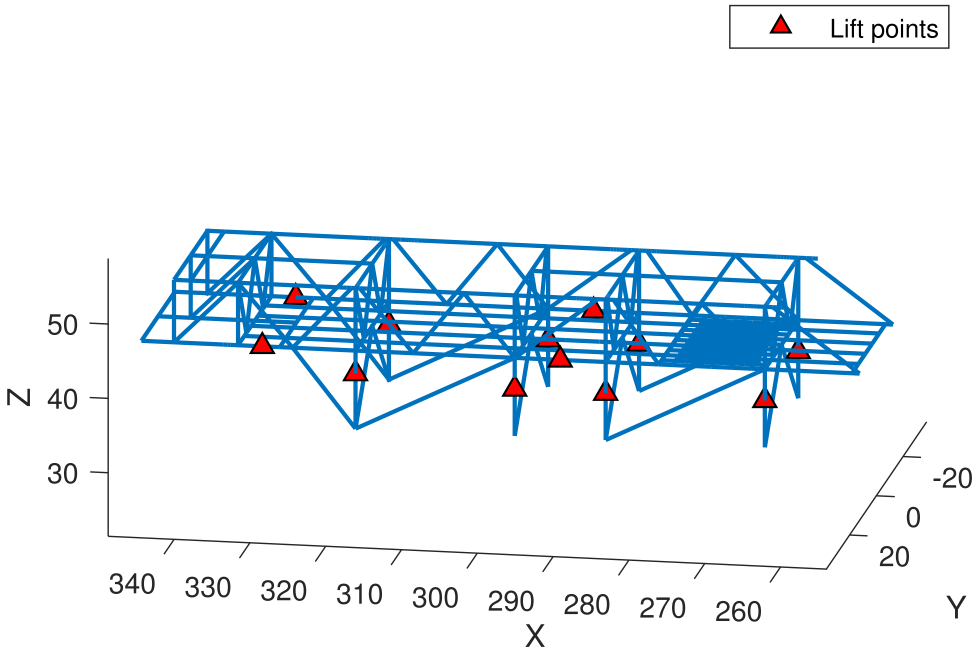


Figure 3.3: Members which are checked for structural integrity and locations of lift points

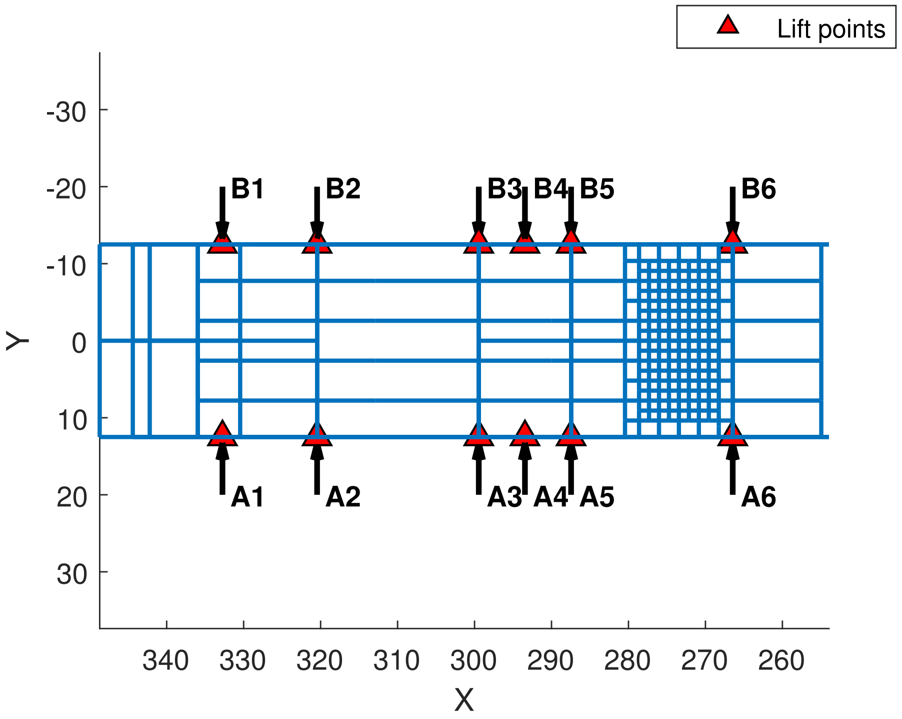


Figure 3.4: Top view of MSF with numbering of lift points

3.2. Implementation of the boundary conditions into the algorithm

In eqs. (2.53) to (2.58) 6 different boundary conditions were stated, and 5 of them will be implemented according to eq. (1.1). Each row of matrix $\boldsymbol{A}$ represents one constraint, and the maximum value of that constraint is stated in the corresponding row of vector $\boldsymbol{b}$. Matrix $\boldsymbol{A}$ is multiplied with the first half of eq. (2.50), containing the integer values which decide whether a lift point is used or not. The first 2 rows of matrix $\boldsymbol{A}$ in eq. (3.2), is the restriction that a maximum of four forklift units of lifting beams can be used per side, according to the definition in eqs. (2.53) to (2.54) and the description of the TLS in section 1.2. Row three and four represent eqs. (2.55) to (2.56), and make sure a minimum of two lift points is used on each side of the topsides.

Furthermore, there are some geometric constraints, which are implemented according to eq. (2.58). Underbeam lift points A4 & B4 can not be used together with the leg lift points next to them (A3, A5, B3 & B5), as the lifting beams do no fit that close to each other. These constraints are stated in the four lowest rows of matrix $\boldsymbol{A}$.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} A1 \\ A2 \\ A3 \\ A4 \\ A5 \\ A6 \\ B1 \\ B2 \\ B3 \\ B4 \\ B5 \\ B6 \end{bmatrix} \leqslant \begin{bmatrix} 4 \\ 4 \\ -2 \\ -2 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{3.2}$$

The reaction forces at the lift points should be lower than the capacity of the TLS (section 2.5 and eq. (2.57)). For calculating the capacity of the TLS, a serviceability limit state (SLS) is used [14]. This means that the self weight is multiplied with weight contingency factor (F_{weight}), CoG variance factor (F_{CoG}), a dynamic lift factor (F_{dyn}) and a SLS factor (F_{SLS}). The values for these safety factors can be found in table C.5. As the removal operation will be done on dynamic positioning (DP), the vessel does not stay on the exact same position. To overcome this problem, a maximum DP footprint has been calculated. This maximum footprint is equal to 0.5 meter and is deducted from the absolute distance from the center line of the vessel. Figure 2.14 is then used to calculate the capacity of every lift point.

A nonlinear inequality function is implemented to incorporate the TLS capacity constraints. This inequality function operates according to eq. (3.2). The genetic algorithm provides a possible lift configuration. The FEM procedure is run for this lift configuration and the reaction forces at each lift point are calculated. Each reaction forces should be lower than the corresponding capacity, otherwise a penalty value is introduced according to eq. (2.59).

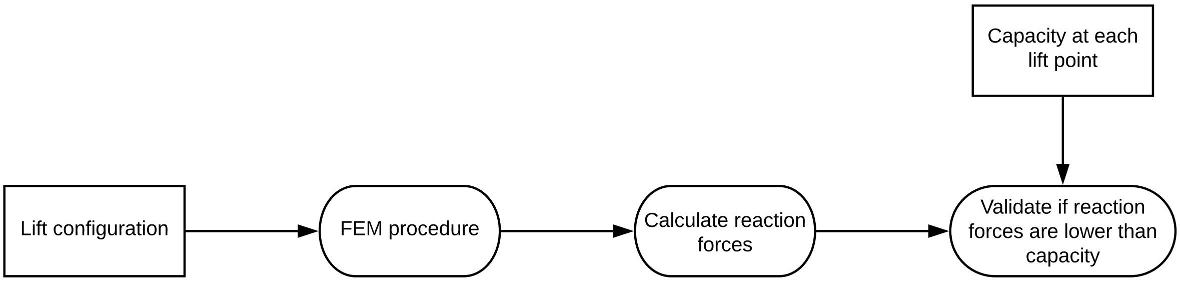

Capacity at each lift point
Lift configuration
FEM procedure
Calculate reaction forces
Validate if reaction forces are lower than capacity

Figure 3.5: Implementation of the nonlinear inequality function incorporating the TLS capacity constraints

3.3. Current lift configuration

In order to decide if the algorithm comes up with good solutions, it is necessary to have a reference case. As there is already a plan on how to lift this topside, this plan will be used. The reference case has 8 on-leg lift points, as shown in fig. 3.6. The maximum utilizations are also visible in this figure, where for clarity only utilization of 0.5 or higher are shown. The corresponding reaction forces can be found in table 3.4. According to the cost function in section 2.4, this lift configuration has a total cost of 2.19 million euros.

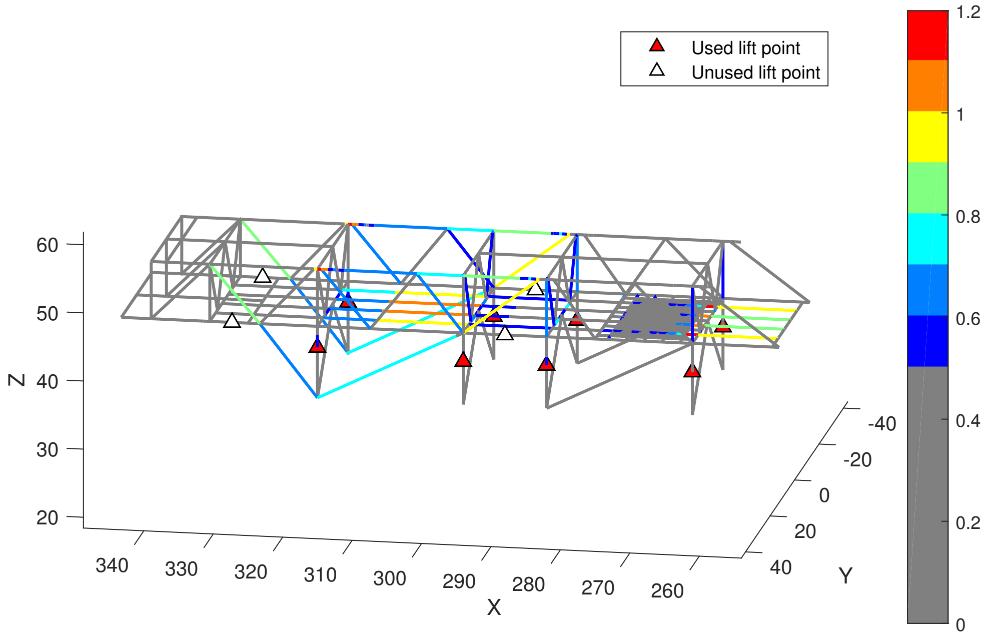


Figure 3.6: Maximum utilizations of the reference case with a cost of 2.19 million euros

Table 3.4: Reaction forces and heights of the solution of 2.19 million euros

Lift points	A2	A3	A5	A6	B2	B3	B5	B6
Relative height [m]	-0.02	0.02	0.01	0	-0.02	0.02	0.01	0
Reaction Force [MN]	43.0	19.8	24.3	30.8	42.8	20.2	25.0	31.4

In order to have a more detailed look at the results, the element numbers of the beams which need reinforcements are shown in fig. 3.7. The corresponding element numbers are also found in table 3.5, where detailed information about each beam is shown. The type of failure is specified, where in case of a combined failure, it is indicated whether buckling happens according to the definition in section 2.4.2.

It is clearly visible, that the 2 longest beams in the middle of the structure (698 & 699) cause the majority of investments needed for reinforcements, which is logical as those beams are significantly longer than the other beams. These two beams fail because of a combined loading, mainly caused by a bending moment in those beams. The bending moment is mainly induced from the reaction forces of lift points A3 & B3.

The four small beams on the right side of the structure (656, 661, 666 & 671) all fail caused by a combined failure with buckling. The bending moment in these beams is quite high, which is caused by the loading of lift points A6 & B6. Contrary to the members closer to these lift points, these members fail. The is caused by the dimensions of these members, which are quite a lot smaller than the members closer to the lift points.

On the left side of the platform, four members fail (80, 81, 83 & 84). Two are located next to each other on both sides. On each side, one member fails with buckling, while the other member fails without buckling. The members which have a buckling failure, have a higher cost. This is logical, as for these members a longer length is taken, according to section 2.4.2. The reason for locating two beams so close to each other, is the difference in dimension between the beams. In reality, this is a joint failure, but these are not checked separately.

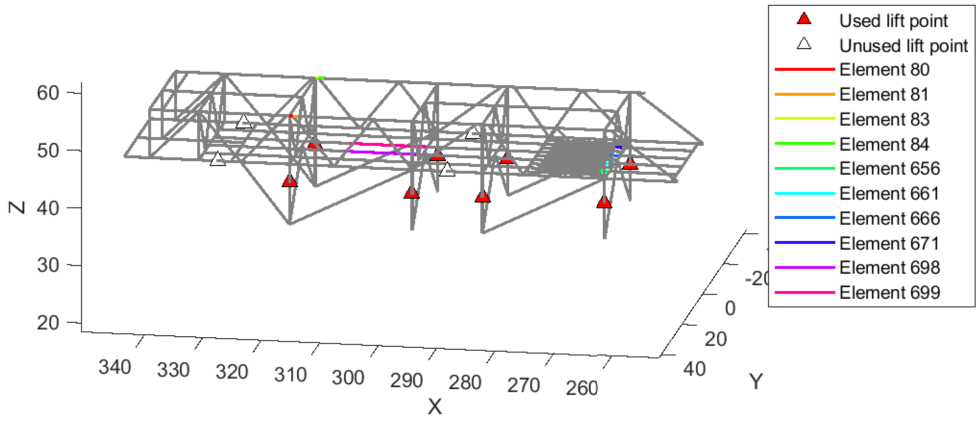


Figure 3.7: Numbering of elements which have an utilization higher than 1.0 of the solution of 2.19 million euros

Table 3.5: Cost and failure data on members which need reinforcements

Element number	Element type	Utilization [-]	Type of failure	Cost [€]
80	Rectangular tube	1.14	Combined (no buckling)	5,962
81	Rectangular tube	1.07	Combined (buckling)	13,344
83	Rectangular tube	1.14	Combined (no buckling)	5,962
84	Rectangular tube	1.08	Combined (buckling)	13,344
656	I-beam	1.14	Combined (buckling)	15,319
661	I-beam	1.02	Combined (buckling)	5,000
666	I-beam	1.02	Combined (buckling)	5,000
671	I-beam	1.12	Combined (buckling)	12,978
698	I-beam	1.05	Combined (no buckling)	46,195
699	I-beam	1.07	Combined (no buckling)	67,088

4

Analysis and improvement of the optimization process

This chapter is divided in two parts. First an analysis on the behavior of the cost function is done in section 4.1. Next, the performance of the algorithm is improved in section 4.2.

4.1. Analysis on the behavior of the cost function

To get more insight in what is happening within the optimization process, a lot of simulations have been run. The found optimal lift configurations have been compared with each other, and a remarkable resemblance is discovered between the different optimal lift configurations. Most of the optimal lift configurations are formed out of either one or a combination of bending and twisting of the topsides. Section 4.1.1 explains the definition of bending and twisting of the topsides. Next, section 4.1.2 shows the influence of bending and twisting on the value of a lift configuration. Finally, section 4.1.3 explains what the effects of increasing the capacity of TLS would mean for finding an optimal solution.

4.1.1. Definition of bending and twist

The bending and twisting behavior is simulated for two different lift configurations: a lift configuration which uses all 8 on-leg lift points and a lift configuration which uses 6 on-leg lift points (A2, A5, A6, B2, B5 & B6). Also combinations of bending and twisting have been simulated, which are a summation of the individual effects. Figure 4.1 contains two schematic drawings of the deck level of a topsides, in which the lift points are shown.

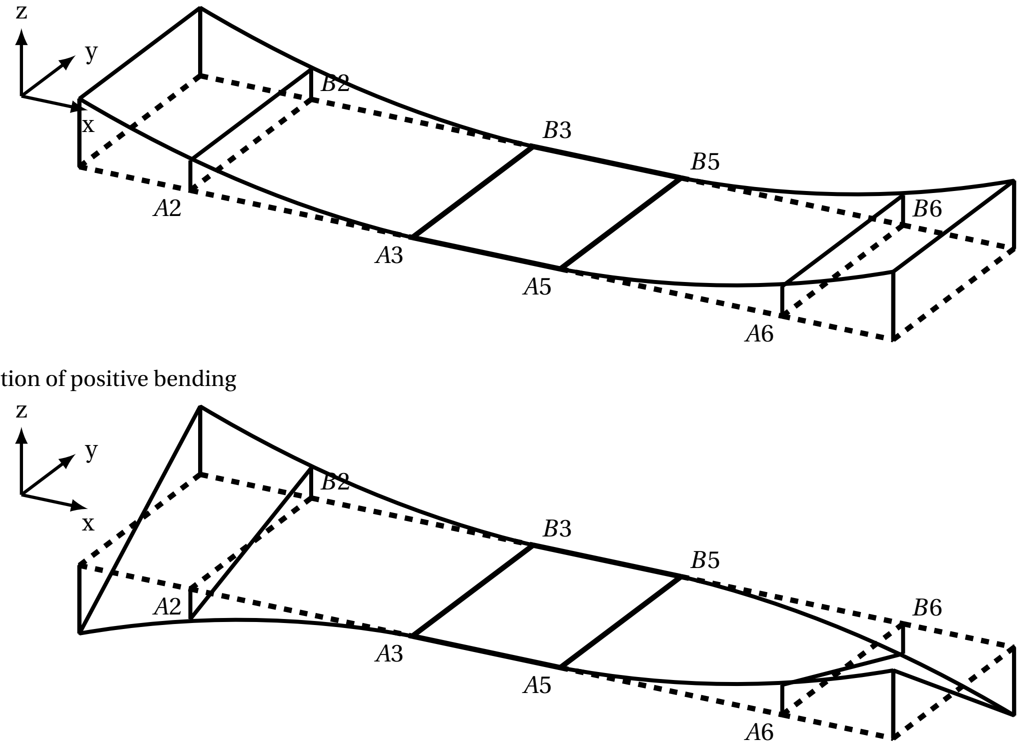
Figure 4.1a shows the definition of positive bending. For the 8 lift point configuration, the middle 4 lift points (A3, A5, B3 & B5) are fixed, while the lift points on the ends of the MSF are all moved up with the same height. For the 6 lift point configuration, the two lift points in the middle (A5 & B5) are fixed, while the other lift points are displaced with the same height.

Figure 4.1b shows the definition for positive twist. For both the 6 and the 8 lift points configurations the middle lift points are fixed again. For positive twist, lift points B2 & A6 are moved up, while lift points A2 & B6 are moved down the same height simultaneously.

4.1.2. Effects of bending and twist on the value of a lift configuration

Figure 4.2a shows the effect of adding bending and twist to the 8 lift point configuration. The grey area is the unfeasible area, where at least one of the lift points has a higher load than the capacity of the TLS (section 2.5). A high amount of either bending or twist leads to a high cost. This is logical, as a high amount of either one of these causes an increase in the stresses in the topsides. The minimum is located at a small area around a bending moment of zero. The twist is a little bit negative. This negative twist is added to reduce the loads on leg B2, in order to get to a feasible solution.

In fig. 4.2b the effects of bending and twist is shown for the 6 lift point configuration. For lower values of twist, a small amount of bending does not make much difference. For higher (absolute) values of twist (more than 0.01 meter), positive bending has a lower cost than the same amount of negative bending. The total



(a) Definition of positive bending

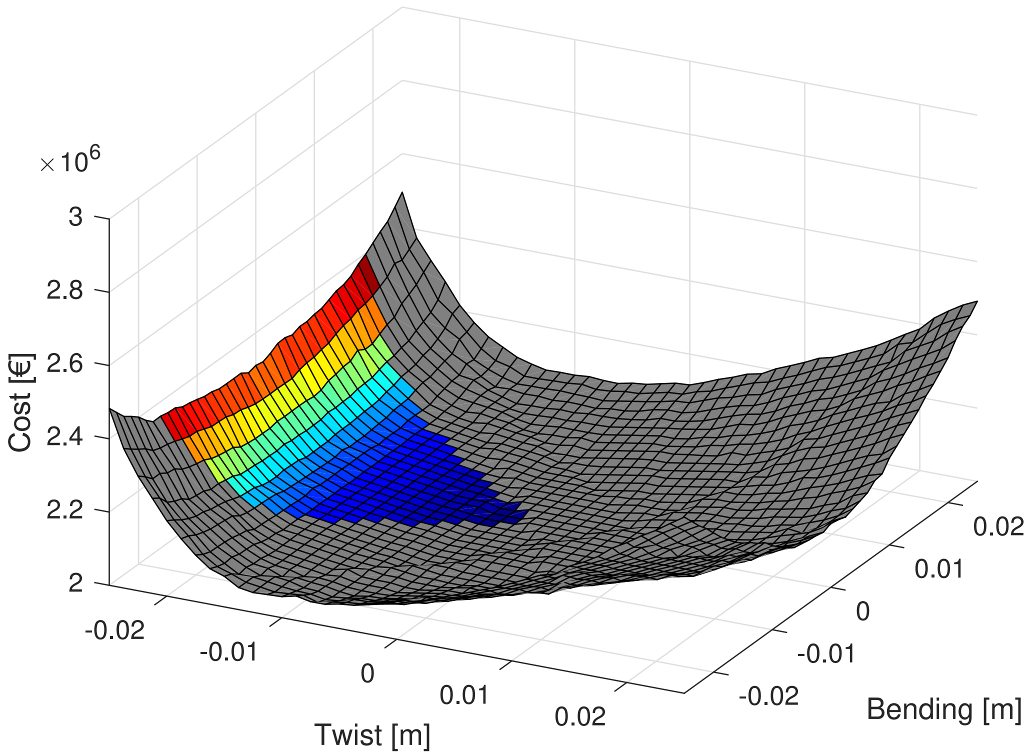
(b) Definition of positive twist

Figure 4.1: Schematic drawing of a topsides

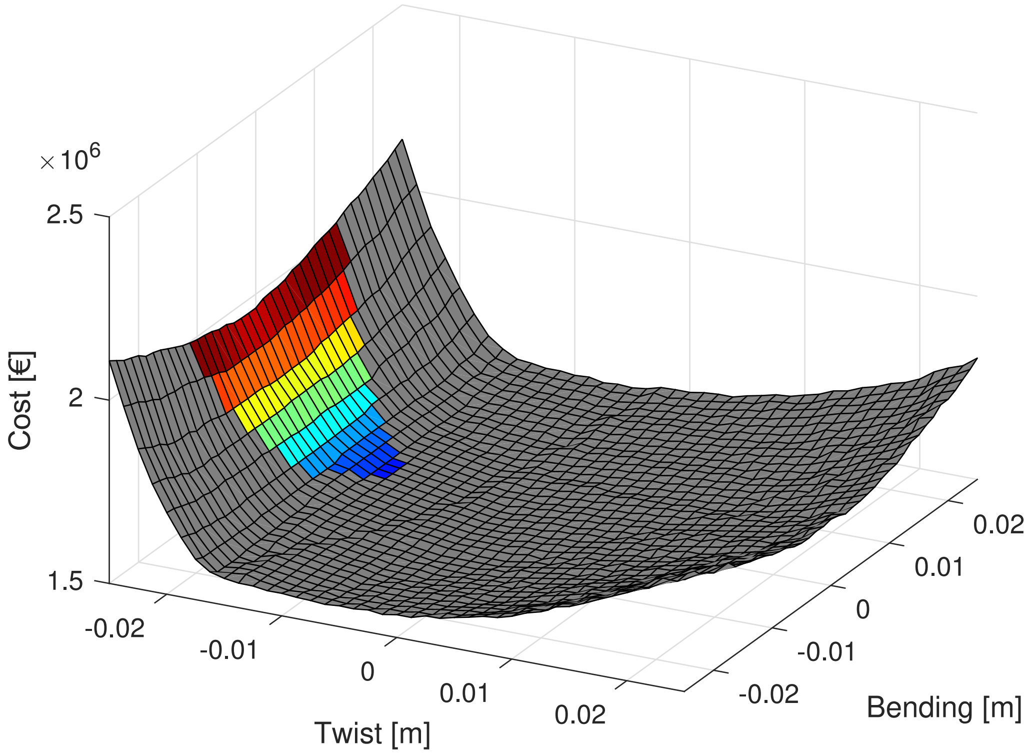
minimum is found at -0.01375 meter of negative twist, with a value of 1.57 million euros. The negative twist makes sure the solution is feasible.

4.1.3. Effects of the capacity of the topsides lift system

A big part of fig. 4.2b is unfeasible, and the optimal lift configuration according to this figure is located at the boundaries of the feasible area. The effect of increasing the capacity of the TLS is shown in fig. 4.3, where the minimum is indicated by a red diamond. Increasing the capacity of the TLS makes it possible to have a feasible solution with a smaller amount of twist, which makes it possible to find a lift configuration with a smaller amount of reinforcements and thus costs. The minimum which is indicated needs a capacity of 51 MN of the TLS.



(a) Showing the cost and feasible region of the 8 lift point configuration, with imposed displacements according to the definitions in fig. 4.1



(b) Showing the cost and feasible region of the 6 lift point configuration, with imposed displacements according to the definitions in fig. 4.1

Figure 4.2: Bending and twisting for a 8 and a 6 lift point configuration

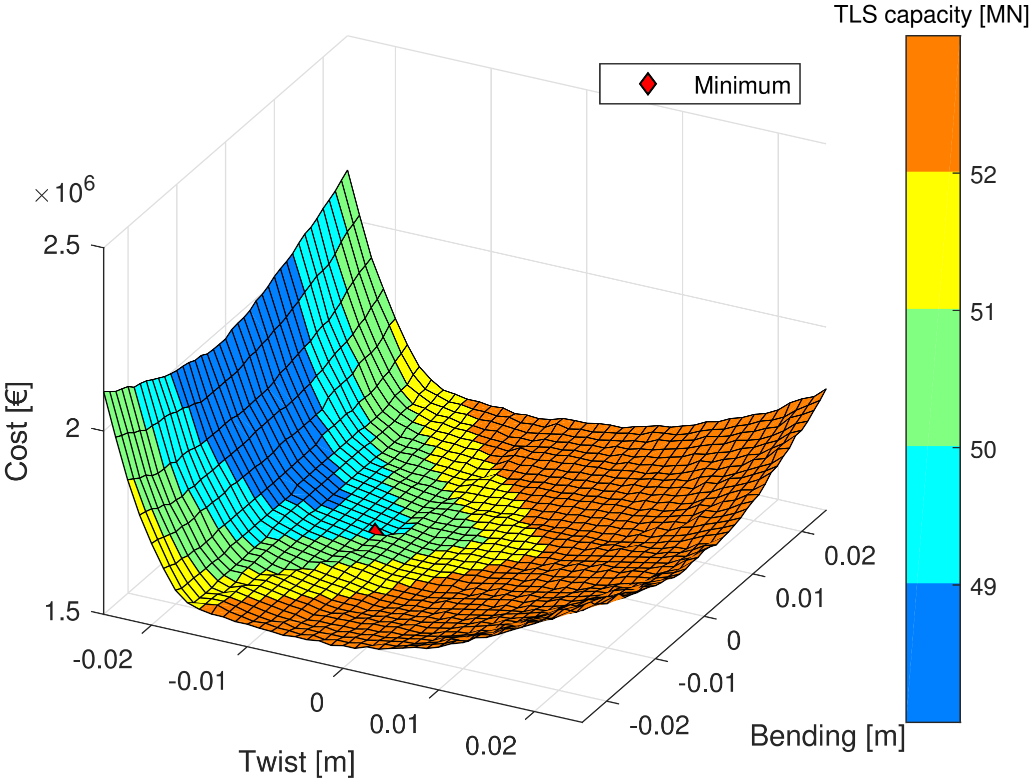


Figure 4.3: Showing the cost and feasible region of the 6 lift point configuration versus the capacity of the TLS, with imposed displacements according to the definitions in fig. 4.1

4.2. Improvement of the performance of the genetic algorithm

To try to optimize the performance of the algorithm, the cross-over fraction and population size have been varied. An adjustment to the algorithm has been made to be able to use a bigger first generation. Furthermore the constraint tolerance has been increased. To be able to the compare different versions of the genetic algorithm, all versions have been run 100 times using the same random seeds. Subsequently the optimization process is shown for each version for the first random seed. The different versions are then compared for success rate, average value and average time. Success has been defined as obtaining a solution with a value equal or lower to 1.6 million euros, in accordance with the lift configuration later provided in section 5.3.

4.2.1. Base case

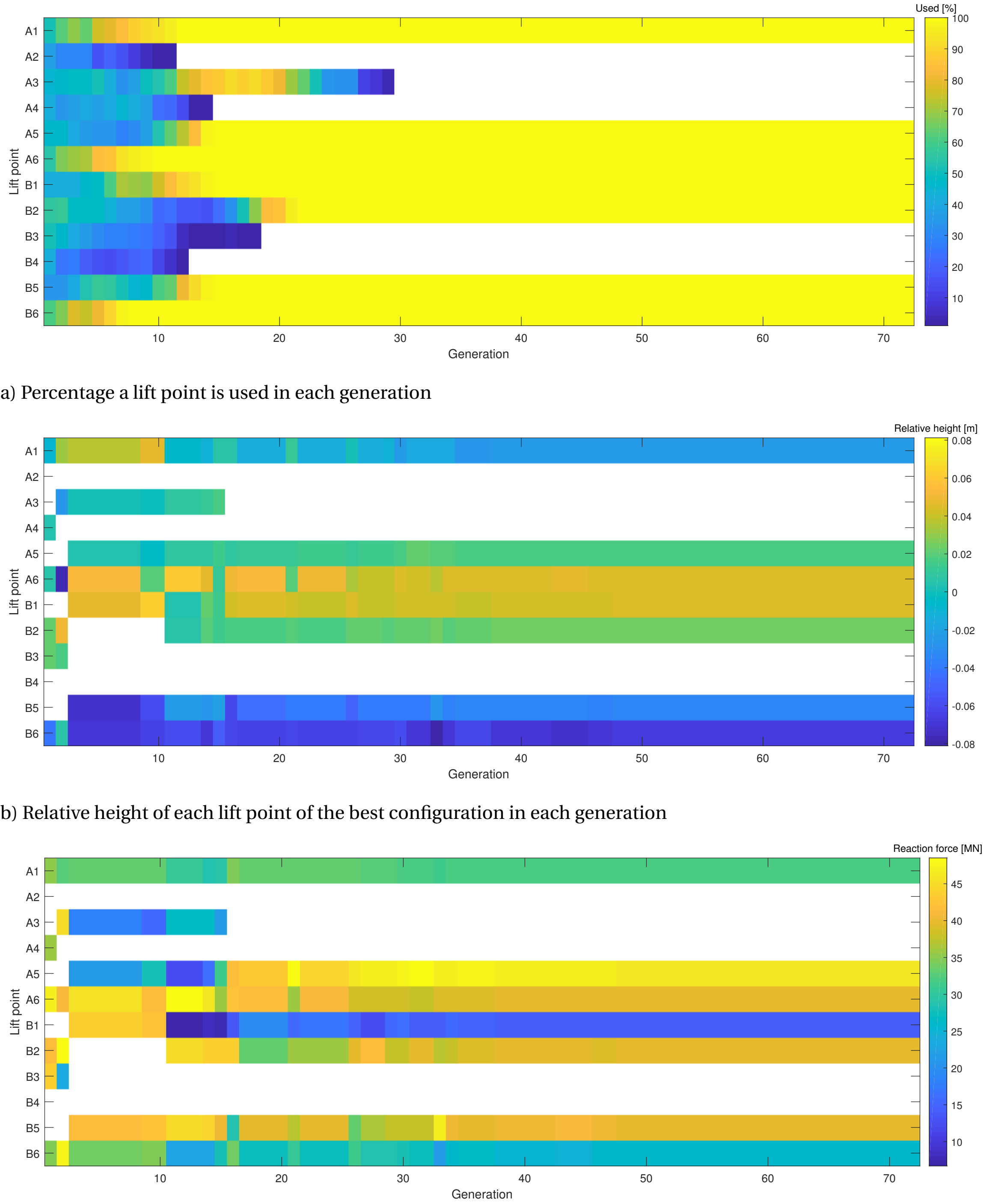
To be able to make a comparison, first a base case has been analyzed. In this base case the standard options from the mixed-integer genetic algorithm from Matlab are used [11], which have a crossover-rate of 0.8, a population size of 100 and a function tolerance of 10^{-6}. The results can be found in table 4.1. The success rate is 24 % and the average time is 420 seconds.

In order to find an explanation for the low success rate, a deeper look has been taken into the optimization process of the algorithm with random seed 1, plotted in fig. 4.4. This optimization process comes to a lift configuration of 1.76 million euros, which is known to be non-optimal. Figure 4.4a shows the percentage a lift point is used in each generation. Quite early in the process lift point A2 is already dropped as a possibility, while in the optimal lift configuration this lift point is used. This is caused by the fact that the lift configurations which use this lift point are not feasible, while other lift configurations are feasible. According to section 2.6.1 feasible solutions are always preferred over unfeasible solutions, which in this case effects in finding a local minimum.

Figure 4.4b shows which lift points are used in the best lift configuration, and the relative height of each lift point. Lift point A3 is used in the beginning in the best lift configuration, and between generation 12 and 20 it is actually used for most lift configurations. However, after generation 15 this lift point is not used anymore in the best lift configuration. Thanks to the mutation algorithm, a new optimal lift configuration has been found. The algorithm converges to this lift configuration, and after generation 30 only this configuration is considered. The relative height is still adjusted to find a better distribution of forces, which is also visible in fig. 4.4c.

Table 4.1: Results from the original setting from the GA

	Value [€]	Generations [-]	Function count [-]	Time [s]
Average	1,906,011	118	11,934	420
Best	1,569,892	60	6,101	170
Worst	2,889,444	386	38,701	1,642



(a) Percentage a lift point is used in each generation

(b) Relative height of each lift point of the best configuration in each generation

(c) Reaction force of each lift point of the best configuration

Figure 4.4: The evolution of a lift configuration over different generations using the standard settings of the GA for random seed 1

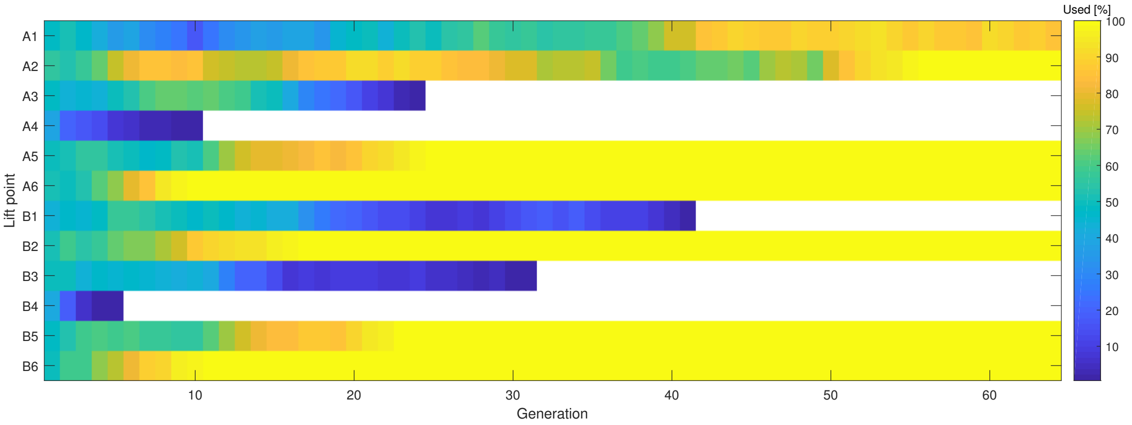
4.2.2. Population size

The standard population size in the reference case is 100. In section 4.2.1 a local optimum was found in 76 % of the runs of the algorithm. An increased population size increases the probability of finding a global optimum solution, as more solutions are tried per generation. The probability of skipping a local or global minimum will be lower. This can be seen in the results (table 4.2), as the average value is 6 % lower than in table 4.1. The optimal solution has now been found 40 times, which is more than 50 % higher than the base case. The drawback of increasing the population size, is that is also increases the total amount of function evaluations quite significantly (and thus increases the time it takes to run the algorithm).

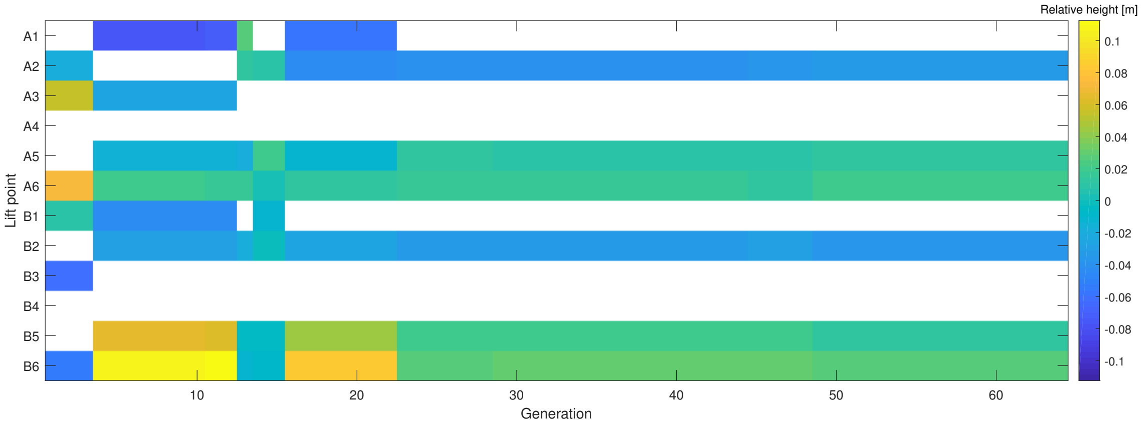
To check if the optimization behavior is actually different from section 4.2.1, the evolution of a lift configuration using a population size of 200 is shown in fig. 4.5. This optimization run does actually find a global optimum, with a value of 1.56 million euros. First of all, in fig. 4.5a it is visible that the search space of the algorithm stays bigger in the beginning. Actually, even in the end two different lift configurations are still tried out. Furthermore, in the first 25 generations in fig. 4.5b, it is visible that each time when a new best lift configuration is found, the new best solution actually does not originate from the former best solution. Because of a bigger population size, the best solutions tends to dominate the population in a slower pace (fig. 4.5a), than using the original settings (fig. 4.4a). Using a bigger population size also causes a quicker discovery of an optimal force distribution (fig. 4.5c).

Table 4.2: Results with a population size of 200

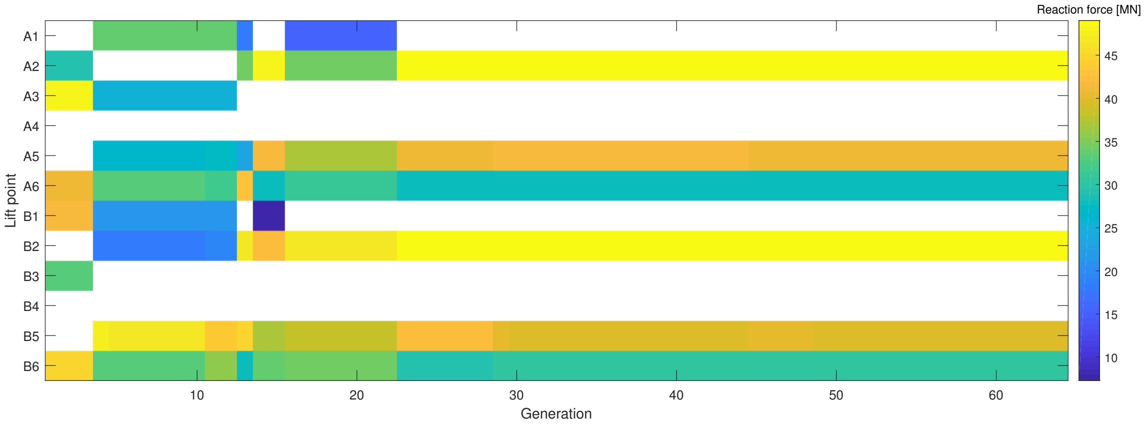
	Value [€]	Generations [-]	Function count [-]	Time [s]
Average	1,789,192	807	25,423	126
Best	1,559,892	208	13,201	65
Worst	2,415,983	2163	53,801	268



(a) Percentage a lift point is used in each generation



(b) Relative height of each lift point of the best configuration in each generation



(c) Reaction force of each lift point of the best configuration

Figure 4.5: The evolution of a lift configuration over different generations using a population size of 200 for random seed 1

4.2.3. Cross-over fraction

To try to improve the performance of the algorithm, the cross-over fraction has been changed as well. A higher cross-over fraction means, that more children are combinations from parents and less children are mutated (and thus the randomness decreases). The standard cross-over fraction is 0.8, results for a cross-over fraction of 0.7 can be found in table 4.3 and results for a cross-over fraction of 0.9 can be found in table 4.4.

The average results with a cross-over fraction of 0.7 are 6% lower than the base case. The minimum of 1.6 million is found in 13 of 100 times, while in the base case this was already achieved 24 times. The average amount of generations is similar. Increasing the cross-over fraction to 0.9 gives similar results as the base case.

Beforehand, it was expected that decreasing the crossover rate, and thus increasing the mutation rate (and hence increasing the amount of randomness), would increase the search area, and in this way would improve the results. In practice the results do not reflect this. Comparing two test runs, fig. 4.6a and fig. 4.7a, it is visible that actually the reverse is happening. The results for a cross-over rate of 0.7 converge faster than the results with a cross-over rate of 0.9.

The same behavior is visible in the force distribution in fig. 4.6a and fig. 4.7a. The best lift configuration changes more with a cross-over rate of 0.7, however, the final optimum found for this cross-over rate has a cost of 1.78 million euros, while the optimum with a crossover rate of 0.9, has a cost of 1.7 million euros.

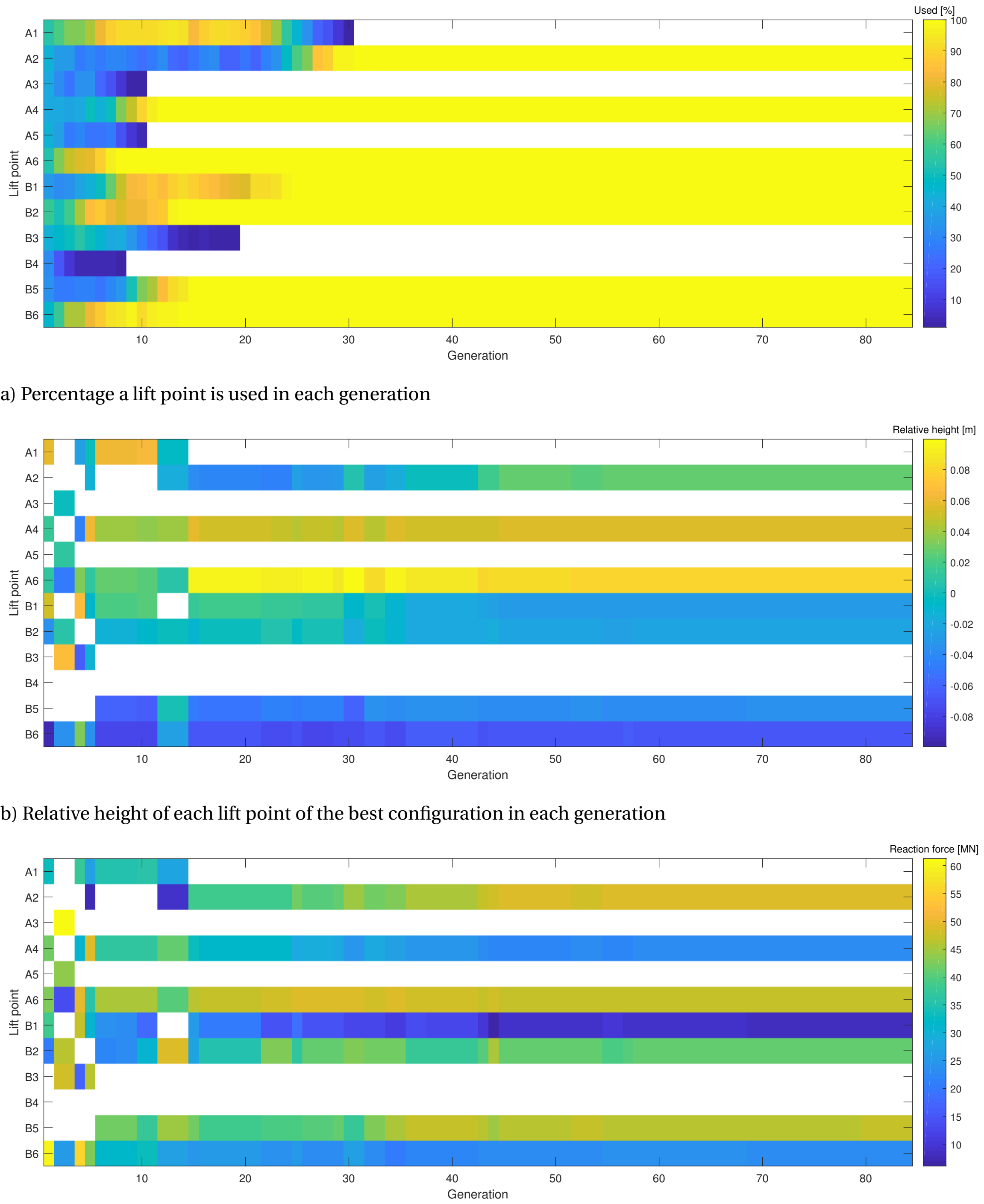
There are two explanations for this. First of all, in the implementation of the cross-over and mutation algorithm (sections 2.6.3 to 2.6.4) the rounding of numbers for integer variables is done randomly. This is a reasonable implementation if the range of the integer values is big, but for integer values which can only be zero or one, this leads to a pure randomness. Accordingly, this randomness will cause many random children with a higher percentage of randomness than the percentage which is tried to achieve. Secondly, too much randomness actually creates many new solutions, of which a very high percentage is unfeasible. As feasible solutions are preferred over unfeasible solution, this randomness does not help for finding a solution.

Table 4.3: Results with a crossover fraction of 0.7

	Value [€]	Generations [-]	Function count [-]	Time [s]
Average	2,015,447	413	12,391	123
Best	1,600,000	170	5,701	56
Worst	2,998,095	1347	37,001	369

Table 4.4: Results with a cross-over fraction of 0.9

	Value [€]	Generations [-]	Function count [-]	Time [s]
Average	1,904,996	413	12,334	122
Best	1,559,892	158	6,601	65
Worst	2,867,104	336	33,701	336

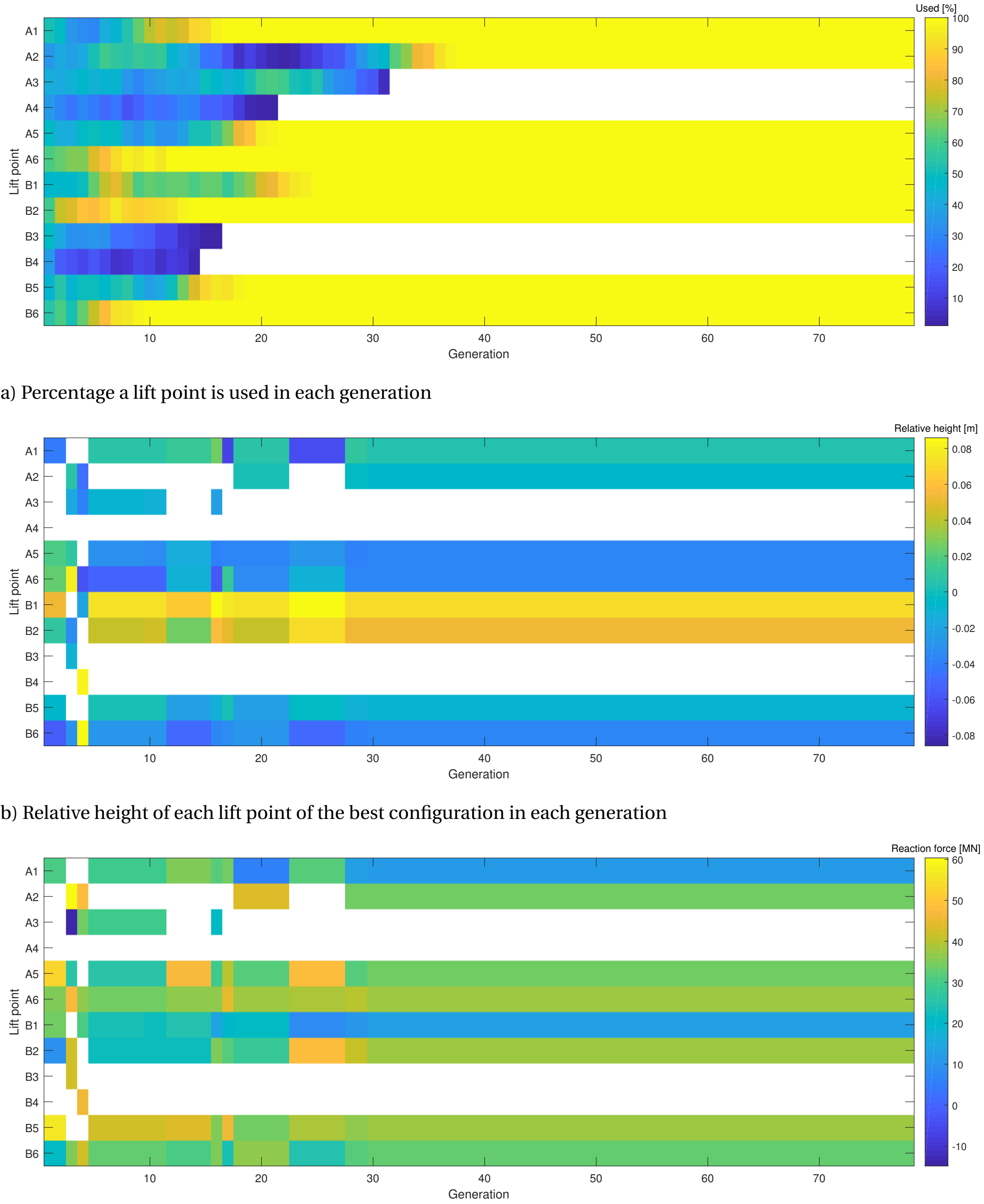


(a) Percentage a lift point is used in each generation

(b) Relative height of each lift point of the best configuration in each generation

(c) Reaction force of each lift point of the best configuration

Figure 4.6: The evolution of a lift configuration over different generations using a cross-over fraction of 0.7 for random seed 1



(a) Percentage a lift point is used in each generation

(b) Relative height of each lift point of the best configuration in each generation

(c) Reaction force of each lift point of the best configuration

Figure 4.7: The evolution of a lift configuration over different generations using a cross-over fraction of 0.9 for random seed 1

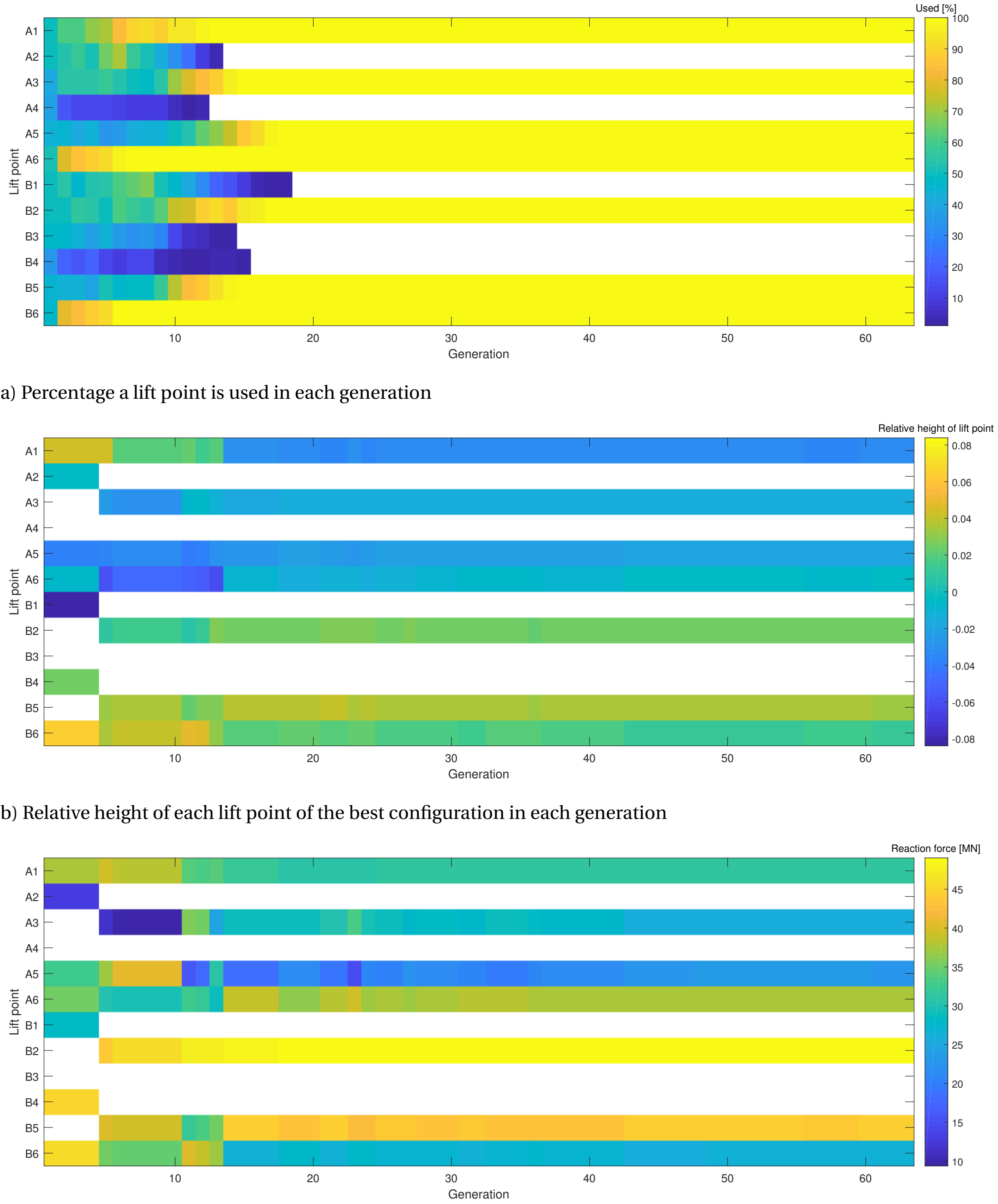
4.2.4. Increased first generation

As feasible solutions are often created using cross-over of other feasible solutions, an increased first generation with a size of 1,000 was added, in order to have a bigger selection pool in the first generation. The best 100 solutions are added to the second generation, and from that moment on the normal GA shall take into force. The results are stated in table 4.5, and the average value has dropped with 10%. In fact, in 79% of the times a global optimum has been found..

Figure 4.8a shows the results for a single run of the GA. In the beginning a bigger diversity is maintained than in fig. 4.4a. Furthermore, multiple feasible lift configurations are considered. From generation 19, only one lift configuration is considered. However, fig. 4.8c shows that the force distribution is still being optimized.

Table 4.5: Results with an increased first generation of 1,000

	Value [€]	Generations [-]	Function count [-]	Time [s]
Average	1,635,481	62	6119	229
Best	1,600,000	51	5201	133
Worst	1,819,303	152	15301	630



(a) Percentage a lift point is used in each generation

(b) Relative height of each lift point of the best configuration in each generation

(c) Reaction force of each lift point of the best configuration

Figure 4.8: The evolution of a lift configuration over different generations using an increased first generation of 1,000 for random seed 1

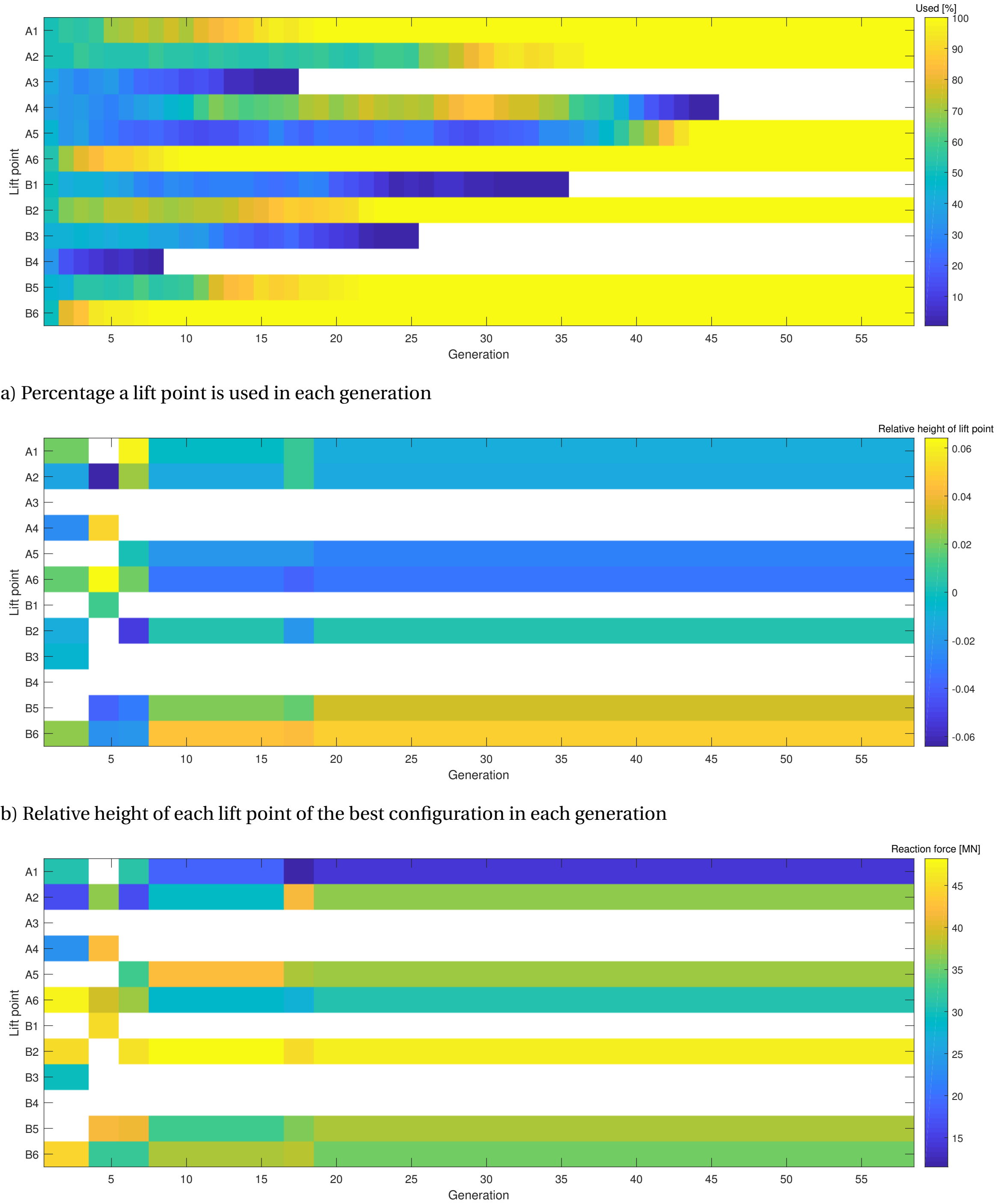
4.2.5. Results on optimization

The final parameters chosen for the parameters are based on the results from sections 4.2.1 to 4.2.4. The population size has been increased from 100 to 200 and the cross-over fraction is changed from 0.8 to 0.9. The first generation is increased to 1,000. The results from these runs are shown in table 4.6, and compared with the genetic algorithm with original settings as provided in section 4.2. The success rate is increased from 24 % to 96 %, while the average time has gone down with 27 % to 304 seconds.

Figure 4.9 shows the changing these parameters, make sure a greater diversity is maintained over different generations. Multiple feasible lift configurations are evolved, ensuring for different combinations of lift points the best force distribution is found. The algorithm can then choose the best solution out of these lift configuration, where table 4.6 proves that this indeed is functioning.

Table 4.6: Results from optimized genetic algorithm

	Value [€]	Generations [-]	Function count [-]	Time [s]
Average	1,607,272	52	20,581	304
Best	1,559,892	51	20,401	56
Worst	1,818,331	72	24,601	435



(a) Percentage a lift point is used in each generation

(b) Relative height of each lift point of the best configuration in each generation

(c) Reaction force of each lift point of the best configuration

Figure 4.9: The evolution of a lift configuration over different generations using the new optimal settings for random seed 1

5 Results on application to topsides removal project

Three possible configurations are discussed and compared with the current lift configuration as discussed in section 3.3. In section 5.1 an improvement is made on a 8 lift point configuration which uses the same lift points as the current lift configuration. In section 5.2 the results are shown of an optimization which only uses leg lift points. Subsequently in section 5.3 the results are shown if also under-beam lift points are used. Finally, section 5.5 states the conclusions on the results.

5.1. Optimization of the 8 lift point configuration

As a lift plan has already been made for a configuration with 8 leg lift points, the easiest change in lift plan is to use lift points, but optimize the height of the lift points to lower the costs of this solution. The algorithm has been run with the restriction that all 8 leg lift point should be used and the results can be found in table 5.1. The improved solution costs 2.006 million, which is a reduction of 8 % from the original configuration as shown in section 3.3. The new utilizations are plotted in fig. 5.1, while data on the 10 elements, which failed in the original configuration (table 3.5), is stated in table 5.2.

The original lift configuration was a convex bow, which is changed to a concave bow, as visualized in fig. 5.2. This change in height of the lift points, lowers the reaction forces in the two middle lift points (A3 & B3), similar to the mass distribution in table 3.3, while the current lift configuration was actually adjusted in order to get a more distributed loading in the lift points. This change in shape is actually not favorable for the topsides.

Changing the shape to a concave bow, lowers the utilization in 9 of the 10 beams which failed earlier, as stated in table 5.2. In fig. 5.1 it is visible that this behavior is present for most of the members op the topsides. The utilization of the two members which originally had the highest cost for reinforcements (698 & 699), is roughly halved, which is caused by both a reduction in the bending moment and the axial loading. The utilization of the four beams on the right side of the structure (656, 661, 666 & 671) is reduced by around 20%, which is caused solely by a reduction of the axial loads.

Table 5.1: Height of lift points & reaction forces of optimized lift point configuration of 2.006 million euros

Lift point	A2	A3	A5	A6	B2	B3	B5	B6
Relative height [m]	-0.02	0	0.01	0.03	-0.01	0	-0.01	-0.01
Reaction force [MN]	48.3	4.1	29.5	36.0	49.0	10.0	31.0	29.5

5.2. Optimization of a lift configuration which only uses on-leg lift points

As a case study has already been conducted on lifting the platform with 8 on-leg lift points, any combination of these 8 would mean no further investigation of the lift points is needed. The algorithm finds a 6 lift point configuration, which can be found in table 5.3. Lift points A2 & B2 are loaded until their maximum capacity, which is to be expected, as the two lift points next to them are not used. It makes sense to not use lift points

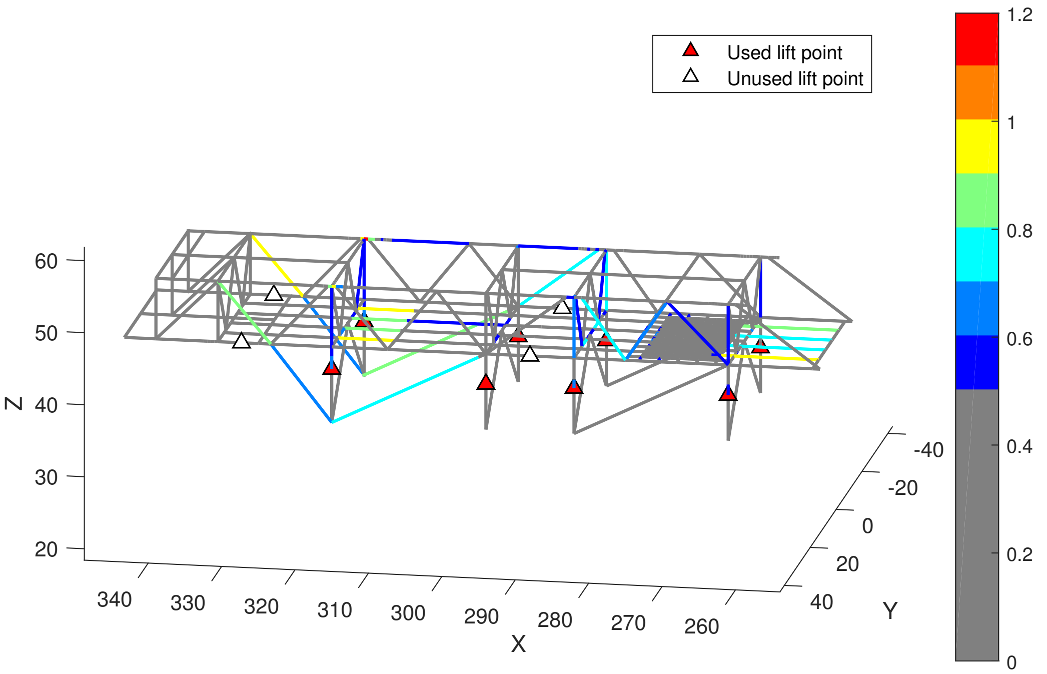


Figure 5.1: Utilizations of the 8 lift point configuration with a cost of 2.006 million euros

Table 5.2: Cost and failure data on optimized lift point configuration of 2.006 million euros and comparison with the original lift configuration

Element No	Element type	Utilization [-]	Ratio [-]	Type of failure	Cost [€]
80	Rectangular tube	0.99	0.88	-	-
81	Rectangular tube	0.69	0.65	-	-
83	Rectangular tube	1.15	1.01	Combined (no buckling)	5,962
84	Rectangular tube	0.87	0.80	-	-
656	I-beam	0.94	0.83	-	-
661	I-beam	0.80	0.78	-	-
666	I-beam	0.78	0.77	-	-
671	I-beam	0.86	0.76	-	-
698	I-beam	0.48	0.46	-	-
699	I-beam	0.57	0.53	-	-

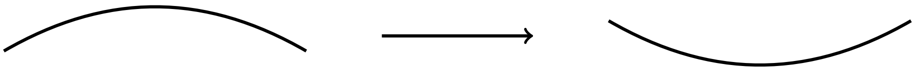


Figure 5.2: Change in shape of the MSF from original lift configuration to optimized lift configuration

A3 & B3, because in the static case (table 3.3) these two legs already have very low loads. The total cost for this solution is 1.56 million, which is 29 % cheaper than the original solution in section 3.3.

Table 5.3: Relative heights and reaction forces of lift configuration of 1.56 million euros

Lift point	A2	A5	A6	B2	B5	B6
Relative height [m]	0.01	0.01	-0.02	0.01	0	-0.02
Reaction force [MN]	49.0	41.0	27.9	49.0	39.5	30.8

In fig. 5.3 the utilizations of this lift configuration are shown. The remarkable difference with original lift configuration, is that, while using less lift points, the amount of members which fail is reduced from 10 to 5. Table 5.4 shows the utilizations of the ten members which failed previously. There are a few reasons why this reduction is achievable. First of all, beams 698 & 699 do not fail anymore since the loads in that region are reduced, which is caused by the removal of lift points A3 & B3. Furthermore, the loads om beams 81 & 83 are also lowered because of the removal of lift points A3 & B3, which causes less load transfers through the beams on the top of the MSF compared to fig. 3.6. Finally, the loads on beams 661 & 666 are not changed significantly, but just enough to prevent failure. However, there is one new beam which fails, with element number 617, which is inflicted by an increase of bending moment caused by a higher loading of lift points A2 & B2.

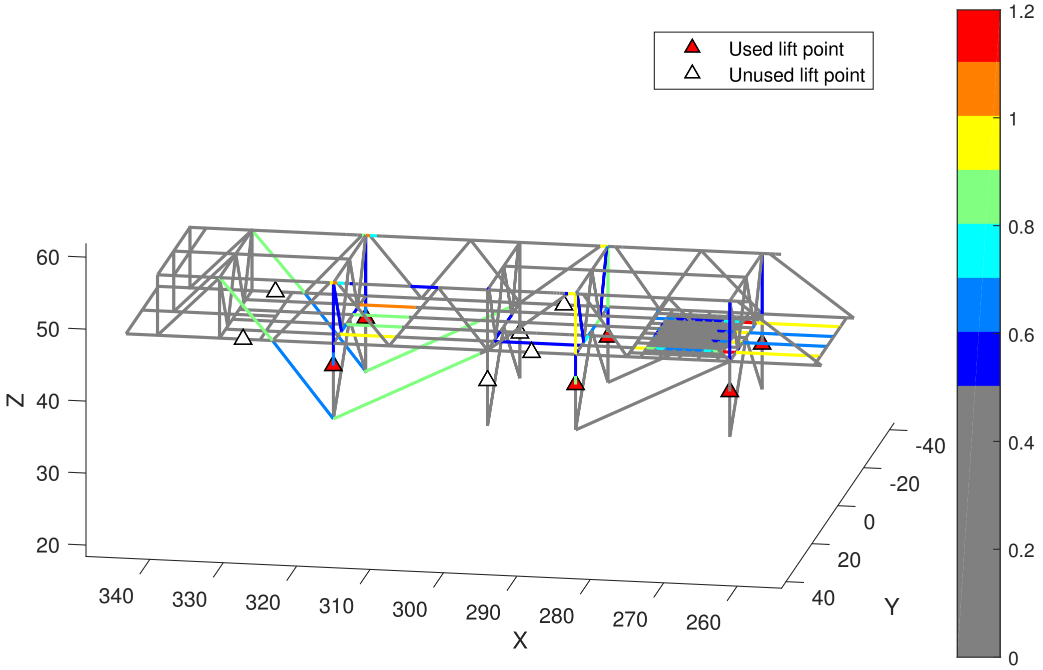


Figure 5.3: Utilizations of the 6 lift point configuration with a cost of 1.56 million euros

5.3. Free optimization of the lift configuration

The final optimization which has been done, is to let the algorithm also use the under-beam lift points. The improved lift configuration can be found in table 5.5. This solution cost 1.6 million euros, which is 27 % cheaper than the original solution and only 3 % more expensive than the solution in section 5.2. However, in this solution not a single beam fails. The cost function has been adjusted to optimize the maximum utilization in stead of the cost of reinforcements, to be able to increase the safety margin to failure of one of the members.

Figure 5.4 shows the utilizations of the final solution. The maximum utilization has been reduced to 0.94 and in total only eight members have a utilization which is higher than 0.8. The utilizations of the ten

Table 5.4: Cost and failure data on optimized lift point configuration of 1.56 million euros and comparison with the original lift configuration

Element No	Element type	Utilization [-]	Ratio [-]	Type of failure	Cost [€]
80	Rectangular tube	1.07	0.95	Combined (no buckling)	5,000
81	Rectangular tube	0.78	0.73	-	-
83	Rectangular tube	1.07	0.93	Combined (no buckling)	5,000
84	Rectangular tube	0.74	0.69	-	-
617	I-beam	1.02	1.39	Combined (no buckling)	10,670
656	I-beam	1.21	1.06	Combined (buckling)	22,342
661	I-beam	0.92	0.90	-	-
666	I-beam	0.90	0.88	-	-
671	I-beam	1.15	1.03	Combined (buckling)	16,880
698	I-beam	0.44	0.42	-	-
699	I-beam	0.45	0.42	-	-

members which failed previously, are stated in table 5.6. The utlizations have gone down significantly. By adding an extra lift point on the left side of the platform, the load on lift points B2 & A2 is reduced and thus leaves a greater safety margin on the capacity of the lifting beams.

Table 5.5: Relative heights and reaction forces of lift configuration of 1.6 million euros

Lift points	A2	A5	A6	B1	B2	B5	B6
Relative height [m]	0.03	0.04	0.05	0.00	0.00	-0.03	-0.07
Reaction force [MN]	47.8	32.3	37.7	13.5	35	45	26

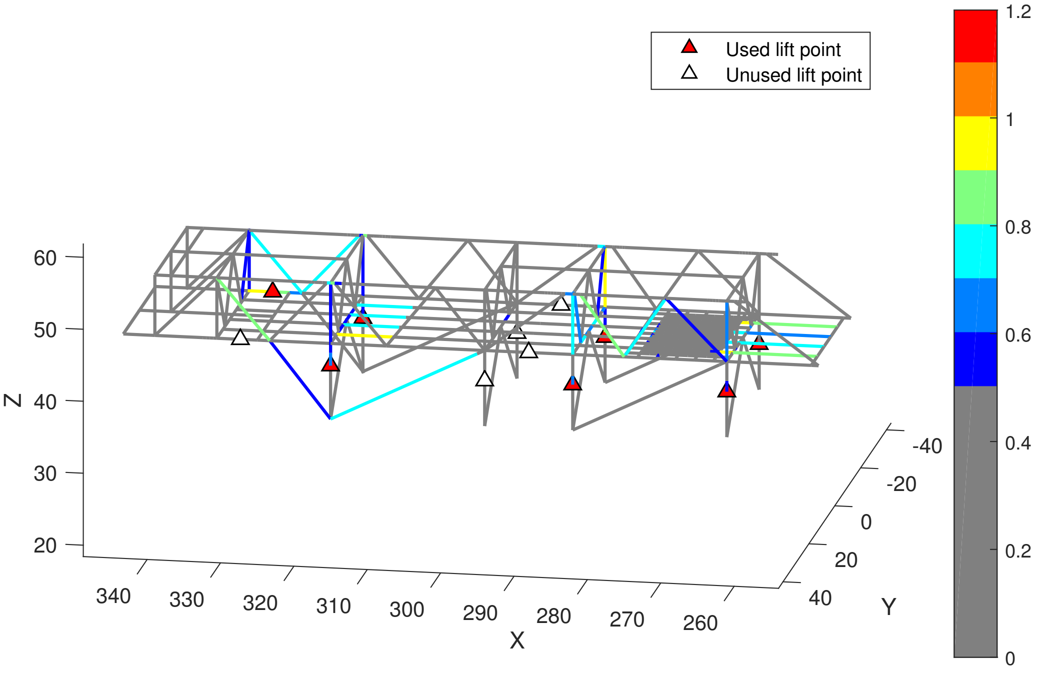


Figure 5.4: Utilizations of the 7 lift point configuration with a cost of 1.6 million euros

Table 5.6: Cost and failure data on optimized lift point configuration of 1.6 million euros and comparison with the original lift configuration

Element No	Element type	Utilization [-]	Ratio [-]	Type of failure	Cost [€]
80	Rectangular tube	0.87	0.77	-	-
81	Rectangular tube	0.55	0.51	-	-
83	Rectangular tube	0.87	0.76	-	-
84	Rectangular tube	0.49	0.46	-	-
656	I-beam	0.94	0.83	-	-
661	I-beam	0.74	0.72	-	-
666	I-beam	0.75	0.73	-	-
671	I-beam	0.91	0.81	-	-
698	I-beam	0.29	0.27	-	-
699	I-beam	0.41	0.38	-	-

5.4. Validation of the results

The optimized results have been validated with the use of FEMAP. The used lift points from the algorithm in chapter 5 are fixed at the same height in FEMAP. The results can be found in table 5.7, where for each category the biggest difference is shown. The overall maximum difference is $7.8 \cdot 10^{-4}$ %, which is regarded as negligible.

Table 5.7: Validation of the results of the algorithm with the use of FEMAP

	Maximum difference [%]
Displacements of free lift points	$1.0 \cdot 10^{-5}$
Reaction forces at lift points	$1.7 \cdot 10^{-4}$
Stresses in elements	$7.8 \cdot 10^{-4}$

5.5. Conclusions

In section 5.1 an optimized 8 lift point configuration is presented. However, this solution is not preferred as both the 6 lift point configuration presented in section 5.2 and the 7 lift point configuration presented in section 5.3 are at least 400,000 € cheaper than the 8 lift point configuration. The 6 and 7 lift point configuration do not differ significantly in cost, but the 7 lift point configuration does not have a single beam which fails. Moreover, the 7 lift point configuration has a bigger safety margin on the capacity of the lifting beams and has a lower maximum utilization. For this reason, the 7 lift point configuration is preferred and comes with a cost reduction of 27 %.

6 Conclusions and recommendations

6.1. Conclusions

This research started with the research question *"Can the lift configuration of a topsides be optimized using a genetic algorithm?"*. Throughout this thesis a method is proposed to find an optimal lift configuration and this method is implemented into an algorithm. The case study shows promising results which proof that this method is feasible.

To answer the research question, a set of research objectives was formulated and studied. These research objectives are discussed below:

- *To choose a problem solving strategy which can automatically search for a favourable lift configuration.*

  A genetic algorithm has been implemented and a cost function has been developed. The genetic algorithm can communicate with the FEM procedure, which makes it possible to rate every solution. In this way the value of a solution can be determined automatically, and good solutions can be used to generate an even better solution.

- *To choose the amount of lift points, the locations of lift points and the amount of force applied on each lift point in a systematic way.*

  A genetic algorithm has been implemented,which chooses, between predefined lift points, the amount of lift points and which lift points to use. Moreover the force on a lift point is also determined by the genetic algorithm. A solution is rated according to the cost function, which makes it possible to choose the lift configuration in a systematic way.

- *To implement a structural model, in order to assess how forces and stresses are distributed in a topside.*

  A FEM procedure has been implemented, which uses the stiffness and the mass matrix of a topside. Using this procedure, the forces and stresses in each element are determined. Furthermore, FE model order reduction has been applied, in order to reduce the amount of DOF's. In this way the computational effort is reduced significantly. Without this reduction, this optimization strategy would not be possible in this time, as for an optimization the amount of function evaluations can exceed 10,000.

- *To determine how many reinforcements are needed to make sure the topside stays intact during a single lift-operation, and where these reinforcements are needed.*

  Using the forces and stresses calculated with the FEM procedure, the ISO 19902 and the AISC 360-10 the utilizations of the members of the MSF have been determined. In this way not only the amount of reinforcements needed are quantified, but also the type is qualified. Furthermore, the percentage with which the maximum allowable stress is exceeded, is known.

- *To quantify the cost of different reinforcements to be able to make a substantiated decision between reinforcing the structure and manufacturing more lift points.*

  The cost for different type of lift points has been estimated according to recommendations from within the company. For different type of failures a reinforcement strategy has been developed. Using this strategy the amount of steel needed for each reinforcement is determined. This is used together with a price for offshore steel to estimate the total investment costs. In this way a substantiated decision can be made between adding extra lift points and reinforcing the topside.

- *The proposed method should be conveniently applicable to different topsides, as there are many removal projects in the future.*

  As the method has been developed using the FE model of the Tyra East platform, and the case study has been done on a different platform, the possibility to apply this method to different topsides has already been shown. If a FE model of a topside is available, it is easy to apply this method to the new topside. The biggest challenge lies in identifying possible lift point, as in early stages of a project the most of the time no detailed information about the bottom of the topside is available.

Furthermore based on the application of the algorithm to a topsides removal project, the following conclusions can be drawn:

- The performance of the genetic algorithm has been improved significantly. By changing the crossover fraction, the population size, the constrain tolerance and adding an increased first generation, the success rate has gone up from 24% to 96%, while the average time o find a solution has gone down with 27% to 304 seconds.
- An optimized lift configuration with 7 lift points is presented. This lift configuration is 27% cheaper than the original lift configuration. Moreover, not a single beam fails in this lift configuration and the maximum utilization is lowered to 0.94.

6.2. Recommendations

Although this study has shown the feasibility of applying this algorithm to find a favourable lift configuration for the removal of a topside with a single-lift operation, the application can still be improved and expanded. Some suggestions for further research are:

- Investigate the cost function on already finished projects and current projects, and compare the estimated costs with the real costs. Improve the cost function where needed, while keeping a general approach which can be used for most projects.
- Include joint failure checks in the algorithm. Some really small members next to joints fail sometimes, which is now regarded as a member failure. In reality, this is probably a joint failure, for which a separate reinforcement strategy can be developed.
- Improve the used boundary conditions in the finite element procedure for lift points. Currently, the lift points are modelled as fixations in the z-direction, and hence can only take up loads in the z-direction. In reality however, as a lift point is lifted with the use of a pair of lifting beams, a difference can be made in the amount of reaction force each lifting beam withstands. In this way it is also possible to take a moment in the y-axis. This moment causes a slight change in the loads and stresses in the members of the topside.
- Choose better locations for under beam lift points. Detailed information about the topside was only available in a late stage of the project. Because of this, it seemed like there were only a few possible locations for under beam lift points. In practice, other locations are viable as well. This would make it possible to place under beam lift points at locations where beams are available to distribute the load better into the structure.
- Make it possible to remove modules. Some topsides are too heavy to be able to remove them in a single-lift operation. A great addition to the algorithm would be the possibility to give a price for the removal of different modules. This addition would make it possible for the algorithm to make a decision between removing different modules and reinforcements. A feasible way for achieving this, is to implement different mass and stiffness matrices for the topsides for every possible combination of remaining modules. In this way, the mass and stiffness distribution can be simulated for every possible combination of lift configuration and remaining modules. Subsequently, the algorithm could come up with a lift configuration which is feasible for the Pioneering Spirit.

- The bottom of many topsides consists of a truss structure. However, there are also some topsides which have a hull bottom, like the Yme platform. Currently is is only possible to define discrete locations for lift points. For topsides with a flat bottom however, it would be useful if the algorithm could choose the lift locations anywhere on the bottom. For these topsides it would be recommended to make it possible to change the location of the lift points to a continuous variable.

- Include shell elements in the cost function. The applied method also works for shell elements, however, so far a choice has been made to focus on the MSF of the topsides. If shell (plate) elements are included, all elements can be evaluated using this algorithm.

- Convert the algorithm to a tool which is easy to use. The algorithm potentially could be a valuable instrument for the company, as a fast and cheap solution can be proposed for a single-lift operation. However, to be able to do this, it should be easy to use for other people as well.

- In the algorithm a possibility has been added to scale the costs of reinforcements for different members. This could be used to scale the costs for reinforcements based on the accessibility of a member. This could be done based on the level where a member is located. Reinforcements under the main deck for example are much more costly than reinforcements on the first deck, as under deck members are harder to reach. The amount of scaffolding is much higher for reinforcing under deck members.

Bibliography

[1] AISC Committee on Specifications. Specification for Structural Steel Buildings. *Standard ANSI/AISC 360-10*, June, 2010.

[2] Jin Cheng and Yin Li. Optimization of structural support locations using a hybrid genetic algorithm. In *2013 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, 2013.

[3] Santos Coelho. An efficient particle swarm approach for mixed-integer programming in reliability – redundancy optimization applications. *Reliability Engineering and System Safety*, 94:830–837, 2009.

[4] William Conley. *Computer optimization techniques*. Petrocelli Books, New York, 4th edition, 1980.

[5] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt. *Concepts and applications of finite element analysis*. John Wiley & Sons, fourth edition, 2002.

[6] Lino Costa and Pedro Oliveira. Evolutionary algorithms approach to the solution of mixed integer nonlinear programming problems. *Computer and Chemical Engineering*, 25:257–266, 2001.

[7] Richard W Cottle and Mukund N Thapa. *Linear and Nonlinear Optimization Associate Series Editor*. Springer, New York, 2017.

[8] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Engrg.*, 186:311–338, 2000.

[9] Kusum Deep and Manoj Thakur. A new crossover operator for real coded genetic alforithms. *Applied Mathematics and Computation*, 188:895–911, 2007.

[10] Kusum Deep and Manoj Thakur. A new mutation operator for real coded genetic algorithms. *Applied Mathematics and Computation*, 193:211–230, 2007.

[11] Kusum Deep, Krishna Pratap, M L Kansal, and C Mohan. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2):505–518, 2009.

[12] Det Norske Veritas. Buckling Strength of Plated Structures. *Recommended Practice DNV-RP-C201*, October, 2010.

[13] Det Norske Veritas. Marine Operations, Design and Fabrication. *Offshore Standard DNV-OS-H102*, January, 2012.

[14] Det Norske Veritas. Design of offshore steel structures, general LRFD method. *Offshore Standard DNV-OS-C101*, July, 2015.

[15] Det Norske Veritas. Marine operations during removal of offshore installations. *Recommended Practice DNVGL-RP-N102*, July, 2017.

[16] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, 1989.

[17] David E Goldberg and Kalyanmoy Deb. Analysis of Selection A Comparative Used in Genetic Algorithms Schemes. *Foundations of Genetic Algorithms*, 1(1):69–93, 1991.

[18] RL Haupt. Antenna design with a mixed integer genetic algorithm. *Antennas and Propagation, IEEE Transactions on*, 55(3):577–582, 2007. doi: 10.1109/TAP.2007.891510. URL `http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=4120263`.

[19] David Mautner Himmelblau. *Applied nonlinear programming*. McGraw-Hill Companies, 1972.

[20] John Holland. Adaptation in artificial and natural systems. *Ann Arbor: The University of Michigan Press*, 1975.

[21] International Organization for Standards. Petroleum and natural gas industries — Fixed steel offshore structures. *International Standard ISO 19902*, December, 2007.

[22] Ian Jack. North Sea oil is in its death throes. But the industry has one last grand act left, apr 2017. URL `https://www.theguardian.com/commentisfree/2017/apr/08/north-sea-oil-death-throes-rigs-decommission-industry`.

[23] Hyun Jeung Ko and Gerald W. Evans. A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs. *Computers and Operations Research*, 34(2):346–366, 2007.

[24] Gary R Kocis and Ignacio E Grossmann. Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis. *Industrial & engineering chemistry research*, 27 (8):1407–1421, 1988.

[25] Daryl L. Logan. *A First Course in the Finite Element Method*. Thomson Learning, 2006.

[26] MathWorks. Global Optimization Toolbox User's Guide (R2018a), 2018. URL `https://www.mathworks.com/help/pdf{_}doc/gads/gads{_}tb.pdf`.

[27] M Ohsaki. Genetic algorithm for topology optimization of trusses. *Computers & Structures*, 57(2):219–225, 1995.

[28] OSPAR Commission. OSPAR Commission Ministerial Meeting of the OSPAR Commission Programmes and Measures. (July):22–23, 1998. URL `http://www.ospar.org/documents?v=6877`.

[29] Peter Prasthofer. Offshore production facilities: decommisioning of topside production equipment. In *The Process of Decommissioning and Removing Offshore and Associated Onshore Oil and Gas Facilities*, volume September, pages 38–47, 1997.

[30] R Rikards, H Abramovich, J. Auzins, A. Korjakins, O. Ozonlish, K. Kalnins, and T. Green. Surrogate models for optimum design of stiffened composite shells. *Composite Structures*, 63:243–251, 2004.

[31] Siemens. Superelement User's Guide, 2016.

[32] A Thaitirarot, RC Flicek, DA Hills, and JR Barber. The use of static reduction in the finite element solution of two-dimensional frictional contact problems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 228(9):1474–1487, 2014.

[33] Jan H Vugts. *Handbook of Bottom Founded Offshore Structures: Part 2. Fixed steel structures*, volume 2. Eburon Uitgeverij BV, 2016.

[34] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. The Finite Element Method: Its Basis and Fundamentals, Sixth edition. *International Journal for Numerical Methods in Engineering*, 2005.

[35] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2013.

A

Self-made implementation of the genetic algorithm

A genetic algorithm has been implemented according to the theory described in section 2.6, and the full code can be found in appendix B. A summary of the procedure can be found in fig. A.1. The main procedure from the genetic algorithm is implemented according to the theory in [16]. Elitism is added, where the best 2% of the population is preserved. Crossover is added according to the theory in [9], mutation is added according to [10], and a method to handle integer problem is added according to [11]. For the mutation algorithm, each variable of an individual has a probability of 3.3% of mutatation.

Test results

This implementation has been validated for three example problems supplied by [11]. The same problems have also been tested with the implementation of a GA in Matlab [26]. All problems have been run 100 times to reduce the influence of randomness. The first problem was originally taken from [4] and is given in eq. (A.1). In this problem all variables y_i are integer. The known optimal solution is $(y_1, y_2, y_3, y_4, y_4; f) = (16, 22, 5, 5, 7; 807)$. The results of both genetic algorithms can be found in fig. A.2. In this figure the amount of successful runs within a certain margin from the optimal solution is shown. It can be seen that, although both GA's find quite some solutions within 5% of the optimal solution, the GA from Matlab is significantly more successful.

$$
\begin{aligned}
\min \quad & f(y) = y_1^2 + y_2^2 + 3y_3^2 + 4y_4^2 + 2y_5^2 - 8y_1 - 2y_2 - 3y_3 - y_4 - 2y_5 \\
\text{subject to:} \quad & \\
& y_1 + y_2 + y_3 + y_4 + y_5 \leqslant 400, \\
& y_1 + 2y_2 + 2y_3 + y_4 + 6y_5 \leqslant 800, \\
& 2y_1 + y_2 + 6y_3 \leqslant 200, \\
& y_3 + y_4 + 5y_5 \leqslant 200, \\
& y_1 + y_2 + y_3 + y_4 + y_5 \geqslant 55, \\
& y_1 + y_2 + y_3 + y_4 \geqslant 48, \\
& y_2 + y_4 + y_5 \geqslant 34, \\
& 6y_1 + 7y_5 \geqslant 104, \\
& 0 \leqslant y_i \leqslant 99
\end{aligned}
\tag{A.1}
$$

error margin on op
solution f=807

The GA has also been tested for a simple mixed integer problem given in [24] and found in eq. (A.2). The known optimal solution is $(x, y; f) = (0.5, 1; 2)$. The results can be found in appendix A. It is visible that Matlab's implementation is more successful.

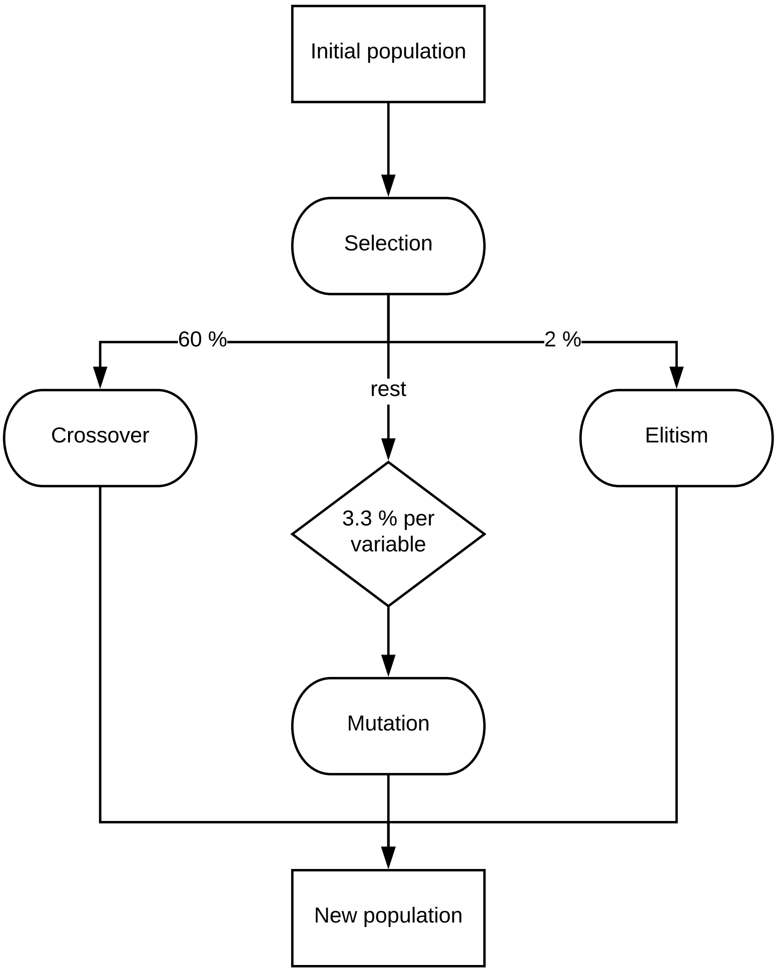


Figure A.1: Procedure of the self-made implementation of a genetic algorithm

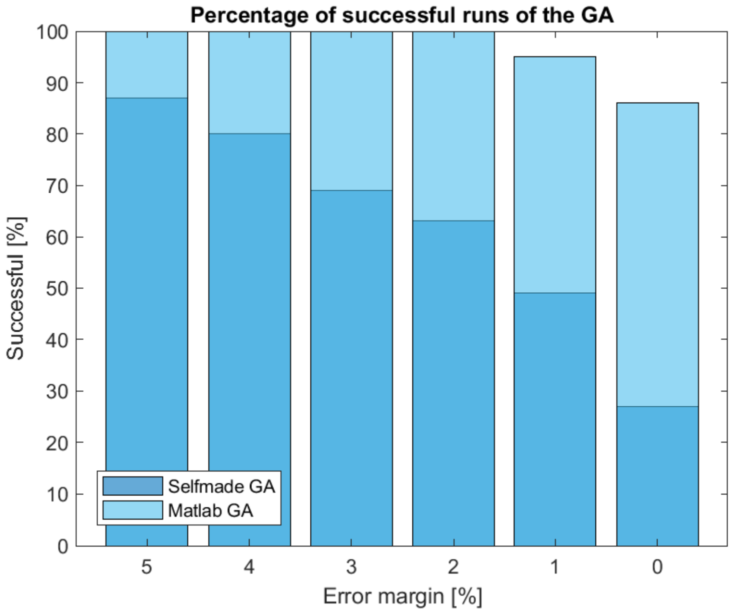


Figure A.2: Error margin on the optimal solution f=807

$$\begin{aligned} \min \quad & f(x,y) = 2x + y, \\ \text{subject to:} & \\ & 1.25 - x^2 - y \leqslant 0, \\ & x + y \leqslant 1.6, \\ & 0 \leqslant x \leqslant 1.6, \\ & y \in \{0,1\} \end{aligned} \tag{A.2}$$

Table A.1: Percentage of successful of the second test problem

	Successful [%]
Selfmade GA	83
Matlab GA	94

Finally the GA has been tested to the problem stated in eq. (A.3), where x_1 is a continuous variable and y_1 & y_2 are integer variables, as defined by [19]. The known local optimal solution is $(x_1, y_1, y_2; f) = (1.5, 50, 25; 0)$, and success is assumed when $f \leqslant 0.01$. Success has been achieved for all 100 runs for both GA's.

$$\begin{aligned} \min \quad & f(x,y) = \sum_{n=1}^{9} [exp(-\frac{(u_n - y_2)^{x_3}}{y_1} - 0.01n]^2, \\ where \quad & u_n = 25 + (-50 log(0.01n))^{2/3}, \\ \text{subject to:} & \\ & 1 \leqslant y_1 \leqslant 100, \\ & 0 \leqslant y_2 \leqslant 25, \\ & 1 \leqslant x_1 \leqslant 5 \end{aligned} \tag{A.3}$$

Conclusions

Matlab's implementation of a genetic algorithm is more successful than the self-made implementation of a GA, and although the exact code of Matlab's GA is unknown, it is expected that the mutation and crossover procedures are slightly different. For this reason it has been chosen to proceed with the GA from Matlab, especially as these test problems are easier than the final problem.

There are a few explanations why Matlab's GA performs better than the self-made GA:

- The core of the self-made genetic algorithm is the theory from Goldberg, introduced in 1989. Although adjustments and improvements have been made, the core is still the old, and probably not best, algorithm.
- The crossover algorithm has the same theory behind the procedure, however, in Matlab's implementation the distance between variables of 2 individuals is different for each variable, while in this implementation is the same.
- The mutation procedure has a probability of 3.3% of mutation per variable. This means that it is very likely that not a single variable mutates. It is very likely MathWorks changed this procedure, to enforce mutation.
- Although the articles on which the implementation of Matlab's genetic algorithm is based, are known, it is very likely that the values for some parameters are different. As it is not possible to find the entire implementation from MathWorks, it is impossible to get the exact same values.

B

Matlab implementation of a genetic algorithm

Initialize procedure

```
%% initialize procedure
% this function sets up the first generation
function [const, stat, oldpop, ncross, nmutation, totmax] ...
    = initialize(data, NASTRAN_DATA)

defining constants  % choosing parameters such as population size,
% number of generation and amount of integer and continuous variables

% initialize counters
nmutation = 0;             % number of mutations
ncross = 0;                % number of crossovers
totmax.worstfitness = 0;   % worst fitness of a generation

% initialize a population at random
for ind1 = 1:const.popsize
    % random initialization of continuous variables
    for ind2 = 1:const.lengthx
        oldpop(ind1).x(ind2) = const.lbreal(ind2) + ...
            (const.ubreal(ind2) - const.lbreal(ind2)) * rand;
    end

    % random initialization of integer variables
    for ind2 = 1:const.lengthy
        oldpop(ind1).y(ind2) = randi([const.lbint(ind2) const.ubint(ind2)],1);
    end

    % defining size of properties of an individual
    oldpop(ind1).feasibility = true;
    oldpop(ind1).parent1 = 0;
    oldpop(ind1).parent2 = 0;
    oldpop(ind1).fitness = 0;
    oldpop(ind1).r = zeros(const.lengthx,1);
    % reaction forces of configuration of a possible solution,
    % calculated in objective function
    oldpop(ind1).FR.superelement = zeros(const.lengthx*3,1);
    % total reaction forces using superposition principle
    oldpop(ind1).FR.total = zeros(const.lengthx*3,1);
    % evaluate initial fitness using objective function
    oldpop(ind1) = objfunc(oldpop(ind1), data, NASTRAN_DATA);

    % saving the worst fitness of this generation
    if oldpop(ind1).feasibility
        if oldpop(ind1).fitness < totmax.worstfitness
            totmax.worstfitness = oldpop(ind1).fitness;
        end
    end
end

% making sure a feasible solution always has a better fitness
% value than a non-feasible solution
for ind1 = 1:const.popsize
    if not(oldpop(ind1).feasibility)
        oldpop(ind1).fitness = oldpop(ind1).fitness + totmax.worstfitness;
    end
end

% saving statistics of this generation and saving the total best solution so far
[stat(1), totmax] = statistics(oldpop, const.popsize, totmax);
end
end
```

Main script

This is the main script, which contains the main loop for generating a new population. This script also retrieves all Nastran data, and calls the statistics procedure.

```
%% starting main script
[data, NASTRAN_DATA] = Combinepunchdatfiles;    % retrieving all Nastran data
gen = 0;             % first generation
% starting the initialize procedure
[const, stat, oldpop, ncross, nmutation, totmax]
    = initialize(data, NASTRAN_DATA);
pop{1}=oldpop;       % saving first population as old generation

% main loop for generating new populations
while gen < const.maxgen
    gen = gen + 1;
    % generating a new population and saving it as newpop
    [newpop, ncross, nmutation] =
        generation(oldpop, ncross, nmutation, const, data, NASTRAN_DATA);

    % saving the worst fitness of a population
    totmax.worstfitness=0;
    for ind1 = 1:const.popsize
        if oldpop(ind1).feasibility
            if oldpop(ind1).fitness < totmax.worstfitness
                totmax.worstfitness = oldpop(ind1).fitness;
            end
        end
    end

    % making sure a feasible solution always has a better fitness
    % value than a non-feasible solution
    for ind1 = 1:const.popsize
        if not(newpop(ind1).feasibility)
                newpop(ind1).fitness = newpop(ind1).fitness + totmax.worstfitness;
        end
    end

    % saving statistics of this generation and saving the total best solution so far
    [stat(gen+1), totmax] = statistics(newpop, const.popsize, totmax);
    oldpop = newpop; % moving current population to old population
end
```

Objective function

```
%% objective function
function gene = objfunc(gene, data, NASTRAN_DATA)

% determining which supports are used (x_fix, 1 = used, 0 = not used)
% determining the height of the used supports
x_height=[0; 0; gene.x(1) * gene.y(1); 0; 0; gene.x(2) * gene.y(2)];
x_fix=[1; 1; gene.y(1); 1; 0; gene.y(2)];
for ind = 3:8
    x_height = [x_height; 0;0;gene.x(ind) * gene.y(ind)];
    x_fix = [x_fix; 0 ;0 ;gene.y(ind)];
end

% system matrices
K = NASTRAN_DATA.Kred;                              % reduced stiffness matrix
g = repmat([0; 0; -9.81], length(gene.x), 1);       % vector of gravity acceleration
R = NASTRAN_DATA.Mred * g;                          % force vector caused by gravity

% temporary K & R matrix used for calculating displacement of free nodes
R2 = R;
K2 = K;

% determining movement of free supports
for ind = 1:length(x_fix)
    if x_fix(ind) == 1
        for ind2 = 1:length(x_fix)
            if ind ~= ind2
                R2(ind2) = R2(ind2) - K2(ind2,ind) * x_height(ind);
            end
        end
        K2(:,ind) = 0;
        K2(ind,:) = 0;
        K2(ind,ind) = 1;
        R2(ind) = x_height(ind);
    end
end

% save displacements and reaction forces
% calculating total reaction forces using superposition principle
gene.r= K2 \ R2;                        % total movement of all retained nodes
gene.FR.superelement = K * gene.r;      % reaction forces caused by this solution
gene.FR.total = F_R_0 + gene.FR.superelement;     % total reaction forces
```

```
% calculating and storing all stresses, checking whether an element fails
for ind = 1:length(data.elements)
% calculating stresses & use of superposition principle
% checking if an element can have stresses
    if not(isempty(data.elements(ind).MES))
        % calculating stress of this solution
        data2.elements(ind).movstress = data.elements(ind).MES * r;
        % use of superposition principle
        data2.elements(ind).totstress = data2.elements(ind).movstress ...
            + data.elements(ind).gravstress;
        % saving total minimum & maximum stress
        data2.elements(ind).maxstress = max(data2.elements(ind).totstress);
        data2.elements(ind).minstress = min(data2.elements(ind).totstress);

        % determining which type an element is and if the element fails
        % use ISO19902 for tubular elements
        if strcmp(data.elements(ind).TYPE ,'TUBE')
            % if tension
            if data2.elements(ind).maxstress > data.elements(ind).SC/gamma_rt
                data2.elements(ind).failure = true;
            % if compression, includes buckling
            elseif data2.elements(ind).minstress <= -data.elements(ind).f_c/gamma_rc
                data2.elements(ind).failure = true;
            else
                data2.elements(ind).failure = false;
            end

        % use Eurocode 3 for I-elements
        elseif strcmp(data.elements(ind).TYPE ,'I')
            % if tension
            if data2.elements(ind).maxstress > data.elements(ind).SC
                data2.elements(ind).failure = true;
            % if compression
            elseif data2.elements(ind).minstress < 0
                if abs(data2.elements(ind).minstress) > data.elements(ind).SC
                    data2.elements(ind).failure = true;
                % check for buckling
                elseif abs(data2.elements(ind).minstress) > ...
                    max(data.elements(ind).S_Rdy, data.elements(ind).S_Rdz)
                    data2.elements(ind).failure = true;
                    fprintf(1,'buckling2')
                end
            else
                data2.elements(ind).failure = false;
            end

        % check for undefined elements
        else
            % if compression
            if data2.elements(ind).minstress < 0 && ...
                abs(data2.elements(ind).minstress) > data.elements(ind).SC
                data2.elements(ind).failure = true;
            % if tension
            elseif data2.elements(ind).maxstress > 250000
                data2.elements(ind).failure = true;
            else
                data2.elements(ind).failure = false;
            end
        end
    end
end

% check if reaction forces are feasible
for ind = 1:8
    if abs(gene.FR.total(ind*3)) > NASTRAN_DATA.grid(ind,5)*1000
        gene.feasibility = false;
        penalty(ind) = -10;
    end
end

% determining fitness value
if gene.feasibility
    y = sum(data2.elements.failure);
else
    gene.fitness = sum(penalty);
end
end
```

Generation

```
%% Generation procedure
% create a new generation through selection, crossover and mutation
% note: generation assumes an even-numbered popsize
function [newpop, ncross, nmutation] = ...
    generation(oldpop, ncross, nmutation, const, data, NASTRAN_DATA)

ind = 3;    % startindex (first 2 are elite solutions)
%vector containing all indices for tournament selection
Tour_vec=repmat(1:const.popsize, 1, 4);
Tour_vec=Tour_vec(randperm(length(Tour_vec)));

% selecting 2 elite solutions
[~,y]=maxk([oldpop.fitness],2);
newpop(1)=oldpop(y(1));
newpop(1).parent1=y(1);
newpop(2)=oldpop(y(2));
newpop(2).parent1=y(2);

% select, crossover and mutation until new population is filled
while ind <= const.popsize
    % pick pair of mates using tournament selection
    mate1 = select_tourn(const.popsize, oldpop, ind, Tour_vec, 1);
    mate2 = select_tourn(const.popsize, oldpop, ind, Tour_vec, 2);

    %crossover and mutation - mutation embedded within crossover
    [newpop(ind), newpop(ind+1), ncross, nmutation] = ...
    crossover(oldpop(mate1), oldpop(mate2), ncross, nmutation, const, mate1, mate2);

    % obtaining fitness value of new population
    newpop(ind) = objfunc(newpop(ind), data, NASTRAN_DATA);
    newpop(ind+1) = objfunc(newpop(ind+1), data, NASTRAN_DATA);

    ind = ind + 2;  % increment population index
end
end
```

Tournament selection

```
%% Tournament selection function
% select a single individual via tournament selection
function ind = select_tourn(popsize, pop, ind4, Tour_vec, matenr)

n_tour = 2;      % number of genes in tournament
fitness=zeros(n_tour, 1);   % vector for all fitness values of tournament
ind2 = 0;
for ind = ((matenr-1)*popsize*2+ind4) : ((matenr-1)*popsize*2+ind4+n_tour-1)
    ind2 = ind2 + 1;
    fitness(ind2) = pop(Tour_vec(ind)).fitness;   %checking fitness of each gene
end

% selecting individual with highest fitness value
[~, ind3] = max(fitness);
ind = Tour_vec(ind3 + (matenr - 1) * popsize * 2 + ind4 - 1);
end
```

Crossover procedure

```
%% Crossover procedure
% cross 2 parent strings, place in 2 child strings
function [child1, child2, ncross, nmutation] = ...
    crossover(parent1, parent2, ncross, nmutation, const, mate1, mate2)

% determing to cross or to mutate
% crossover according to Laplace distribution
if flip2(const.pcross)  % coinflip
    u = rand;
    r = rand;
    ncross = ncross + 2;
    if r <= 0.5
        betareal = const.a - const.breal * log (u);
        betaint = const.a - const.bint * log (u);
    else
        betareal = const.a + const.breal * log (u);
        betaint = const.a + const.bint * log (u);
    end
    child1.x = parent1.x + betareal * abs(parent1.x - parent2.x);
    child2.x = parent2.x + betareal * abs(parent2.x - parent1.x);
    child1.y = parent1.y + betaint * abs(parent1.y - parent2.y);
    child2.y = parent2.y + betaint * abs(parent2.y - parent1.y);


% checking if solution is feasible, otherwise change it for a random value
    for ind = 1:const.lengthx
        if child1.x(ind) > const.ubreal(ind) || child1.x(ind) < const.lbreal(ind)
            child1.x(ind) = (const.ubreal(ind) - ...
                const.lbreal(ind)) * rand + const.lbreal(ind);
        end
        if child2.x(ind) > const.ubreal(ind) || child2.x(ind) < const.lbreal(ind)
            child2.x(ind) = (const.ubreal(ind) - ...
                const.lbreal(ind)) * rand + const.lbreal(ind);
        end
    end

    for ind = 1:const.lengthy
        if child1.y(ind) > const.ubint(ind) || child1.y(ind) < const.lbint(ind)
            child1.y(ind) = randi([const.lbint(ind) const.ubint(ind)],1);
        end
        if child2.y(ind) > const.ubint(ind) || child2.y(ind) < const.lbint(ind)
            child2.y(ind) = randi([const.lbint(ind) const.ubint(ind)],1);
        end
    end

% mutation
else
    child1.x = zeros(1,const.lengthx); child2.x = child1.x;
    for n_cnt = 1:const.lengthx
        [child1.x(n_cnt), nmutation] = ...
            mutation(parent1.x(n_cnt), const, nmutation, n_cnt);
        [child2.x(n_cnt), nmutation] = ...
            mutation(parent2.x(n_cnt), const, nmutation, n_cnt);
    end
    child1.y = zeros(1,const.lengthy); child2.y = child1.y;
    for n_cnt = 1:const.lengthy
        [child1.y(n_cnt), nmutation] = ...
            mutationint(parent1.y(n_cnt), const, nmutation, n_cnt);
        [child2.y(n_cnt), nmutation] =  ...
            mutationint(parent2.y(n_cnt), const, nmutation, n_cnt);
    end
end

% making sure integer variables are integer
for ind = 1:const.lengthy
    if floor(child1.y(ind)) ~= child1.y(ind)
        child1.y(ind) = floor(child1.y(ind)) + randi([0 1]);
    end
    if floor(child2.y(ind)) ~= child2.y(ind)
        child2.y(ind) = floor(child2.y(ind)) + randi([0 1]);
    end
end
end
```

Mutation procedure

```
%% mutation function
% mutate an allele with probability pmutation, count number of mutations
function [allele, nmutation] = mutation(alleleval, const, nmutation, n_cnt)
mutate = flip2(const.pmutation);

% mutation procedure with power distribution
if mutate
```

```
    nmutation = nmutation + 1;
    r = rand;
    t = (alleleval - const.lbreal(n_cnt)) / ...
        (const.ubreal(n_cnt) - const.lbreal(n_cnt));
    s = rand ^ const.preal;
    if t < r
        allele = alleleval - s * (alleleval - const.lbreal(n_cnt));
    else
        allele = alleleval + s * (const.ubreal(n_cnt) - alleleval);
    end
else
    allele = alleleval;
end
end
```

Statistics

```
%% Statistics procedure
% Calculate population statistics
function [stat, totmax] = statistics(pop, popsize, totmax)

% setting values from first individual
stat.sumfitness = pop(1).fitness;
stat.min2 = pop(1).fitness;
stat.max2 = pop(1).fitness;

% calculating sum of fitness & best and worst fitness
for n_cnt = 2:popsize
    stat.sumfitness = stat.sumfitness + pop(n_cnt).fitness; % accumulate fitness sum
    if pop(n_cnt).fitness > stat.max2
        stat.max2 = pop(n_cnt).fitness;   % new max
        % save if a new total best solution has been found
        if stat.max2 > totmax.totalmax && pop(n_cnt).feasibility
            totmax.totalmax = stat.max2;
            totmax.bestx = pop(n_cnt).x;
            totmax.besty = pop(n_cnt).y;
        end

    elseif pop(n_cnt).fitness < stat.min2
        stat.min2 = pop(n_cnt).fitness;   % new min
    end
end
stat.avg = stat.sumfitness / popsize;   % calculate average

end
```

C

Overview of the used safety factors

Resistance of tubular members

For tubular members the resistance against different loads is reduced according to ISO 19902 [21]. The resistance is lowered by dividing the resistance with the corresponding partial resistance factor and the values for all partial resistance factors can be found in table C.1.

Table C.1: Partial resistance factors used for tubular members

Type of loading	symbol	value [-]
Tension	$\gamma_{R,t}$	1.05
Compression	$\gamma_{R,c}$	1.18
Bending	$\gamma_{R,b}$	1.05
Shear	$\gamma_{R,v}$	1.05

Resistance of other members

For tubular members the resistance against different loads is reduced according to AISC 360-10 [1]. The resistance is lowered by multyplying the resistance with the corresponding partial resistance factor and the values for all partial resistance factors can be found in table C.2.

Table C.2: Partial resistance factors used for other members

Type of loading	symbol	value [-]
Tension	$\gamma_{R2,t}$	0.9
Compression	$\gamma_{R2,c}$	0.9
Bending	$\gamma_{R2,b}$	0.9
Shear	$\gamma_{R2,v}$	0.9

Weight of the topside

Uncertainty in the weight of the topside and the location of the CoG is handled according to DNV-OS-H102 [13]. Uncertainty in the dynamic lift loads will be handled with a dynamic load factor, in accordance with other removal project in the company. The weight of the topside will be multiplied with the 3 safety factors found in table C.3.

Table C.3: Weight of the topside

Safety factor	symbol	value [-]
Weight contingency factor	F_{weight}	1.1
CoG variance factor	F_{CoG}	1.1
Dynamic load factor	F_{dyn}	1.15

Capacity of the lifting beams

The capacity of the lifting beams is determined according to fig. 2.14, however, a DP safety margin is deducted from the distance from the center of the topside slot. The loads on the the lifting beams are multiplied with a SLS factor according to DNVGL-OS-C101 [14]. A summary of can be found in table C.4.

Table C.4: Capacity of the lifting beams

Safety factor	value [-]
SLS factor	1.0
DP safety margin	0.5 m

Loads on the members of a topside

Loads on the members of a topside are multiplied with a ULS A factor according to DNV-OS-H102 [13].

Table C.5: Safety factor on the loads on the members of a topside

Safety factor	value [-]
ULS A factor	1.2