

Combined shape and size optimization of an offshore crane

I.C. Vooijs



Combined shape and size optimization of an offshore crane

by

I.C. Vooijs

to obtain the Master of Science degree for the programme Computer Simulations for Science and
Engineering
at the Delft University of Technology,
to be defended publicly on Monday July 27, 2020 at 15:30

Student number:	4360656
Thesis duration:	April 1, 2019 – July 27, 2020
Thesis committee:	Dr. ir. M. B. van Gijzen, TU Delft, supervisor
	Prof. dr. ir. C. Vuik, TU Delft
	Dr. ir. M. Langelaar, TU Delft
	Dr. R. Astudillo, TU Delft
	Ir. A. Sillem, Tetrahedron B.V.

This thesis is confidential and cannot be made public until July 31, 2022.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This thesis is written as part of the graduation project for the Computer Simulations for Science and Engineering (COSSE) master's program. The research was conducted for the company Tetrahedron in the period from November 2019 until July 2020.

Tetrahedron is working on a new design for large offshore cranes that can lift higher. This is necessary due to the increasing size of offshore wind turbines, and the inability of the conventional cranes to scale up without needing a larger ship that is carrying them. To compute the forces working on the trusses of the crane structure, Tetrahedron has developed the software program Automatic Crane Engineer (ACE). Right now the program can already optimize the cross-sectional area of the bars, but the optimization should be extended to include the shape of the structure itself.

The goal of this research is therefore to extend ACE with the necessary functionalities to perform shape optimization of the cranes. The research to achieve this goal is structured in two phases. In the first phase, the literature review part [1], the two research questions have been answered: the formulation of the crane optimization problem is defined and the best optimization methods for this problem have been chosen and implemented in Python. These are the Interior point method as proposed by Byrd, Hribar and Nocedal [2] and the Method of moving asymptotes by Krister Svanberg [3]. As a third method, the Sequential Least Squares Programming method by Dieter Kraft [4] is also tested.

The present report describes the second part of the research. To keep the report self-contained, relevant parts of the literature review report are included. In this report, the selected methods are investigated in more detail and compared with respect to numerical properties and the final results for the crane optimization.

I would like to thank Martin van Gijzen and Reinaldo Astudillo for guiding me the last nine months during this research project. Thanks for traveling all the way to Heijplaat with me, spending many hours meeting about the topic, first physically and later on mostly online due to the corona pandemic. Thank you for all the valuable feedback and always making me feel confident, even though sometimes it was hard to figure out the next steps.

Many thanks to my colleagues from Tetrahedron, especially to Adriaan Sillem. Thank you for helping me understand the mechanics of crane structures, and taking all the time I needed for that. Thank you for all the advice on the report, on the coding, and on the research itself, and thanks for all the meetings, even every day during the final part of the research.

For the past two years I had the chance to study at both the TU Berlin and TU Delft, and I would like to thank the coordinators Reinhard Nabben and Kees Vuik for this opportunity and their guidance during the COSSE program.

I also want to thank Matthijs Langelaar, for being part of my thesis committee and for taking the time to read and value this report.

*I.C. Vooijs
Delft, July 2020*

Contents

List of Symbols	vii
Abbreviations	ix
1 Introduction	1
2 Crane design and conventions	5
2.1 Crane design	5
2.2 Statics of truss structures	6
2.3 Design space	6
3 Mathematical formulation	11
3.1 Direct stiffness method	11
3.2 Objective function	13
3.3 Constraints	13
3.4 Movement of the crane	17
3.5 Bounds	18
4 Nonlinear optimization methods	21
4.1 Sequential Quadratic Programming	22
4.2 Interior-point methods	23
4.3 Convex linearization methods	25
4.4 Method of Moving Asymptotes	27
4.5 Primal-dual solution methods	30
4.6 Comparison of methods	33
5 2D experiments	35
5.1 Optimization methods	36
5.2 Gradient computation	36
5.3 10-bar truss problem	37
5.4 Michell's arch	38
5.5 2D Luffing boom crane	40
5.6 2D Tetrahedron crane	43
6 3D experiment: Tetrahedron crane	47
6.1 Problem definition	47
6.2 Comparison new method with existing optimization method	48
6.2.1 Convergence	51
6.2.2 Robustness	51
6.3 Parameter sensitivity	52
6.3.1 Bar thickness	52
6.3.2 Side-lead	54
7 Discussion and recommendations	57
Bibliography	61

List of Symbols

α	Thickness ratio of annulus	\mathbf{d}_x	Search direction for design variables
$\gamma_{\text{off-lead}}$	Off-lead angle	\mathbf{n}	Joint or node)
$\gamma_{\text{side-lead}}$	Side-lead angle	\mathbf{r}	Residual
λ_b	Internal force of bar b (N)	\mathbf{s}	Slack variable vector
$\mathcal{A}(\mathbf{x}^*)$	Active set of constraint indices	\mathbf{u}	Displacement vector
\mathcal{B}	Set of all bars in the structure	\mathbf{x}	Design variables
\mathcal{L}	Lagrangian function	\mathbf{y}	Auxiliary variables for MMA
\mathcal{N}	Set of all joints	A	Cross-sectional area (m ²)
μ	Barrier parameter	$A_g(\mathbf{x})$	Transpose of Jacobian of inequality constraints
ϕ	Merit function	$A_h(\mathbf{x})$	Transpose of Jacobian of equality constraints
ρ_b	Density of bar b (kg/m ³)	B_k	Approximation of the Hessian
Σ^k	(Approximation of) Hessian of \mathcal{L} w.r.t. \mathbf{s}	$d_{\text{min}}^{\text{boom clearance}}$	Minimal boom clearance (m)
σ_b	Stress in bar b (N m ⁻²)	d^{hoist}	Minimal hoist clearance (m)
$\sigma_{\text{pendants}}^{\text{min}}$	Minimum pendant stress (N m ⁻²)	d_{max}	Maximum safe distance to other legs (m)
σ_{yield}	Yield strength (N m ⁻²)	E	Optimality measure
θ	Angle describing crane movement	E	Young's modulus (N/m ²)
ε_{tol}	Tolerance level parameter	\mathbf{e}	Vector of ones
$\boldsymbol{\alpha}^k$	Lower move limits	\mathbf{F}	Load vector
$\boldsymbol{\beta}^k$	Upper move limits	$f(\mathbf{x})$	Objective function
$\boldsymbol{\eta}$	Lagrange multiplier vector for upper bound inequality of MMA	F_E	Euler's critical load (N)
λ_g	Lagrange multiplier vector of inequality constraints	F_L	Forces on the load node
λ_h	Lagrange multiplier vector of equality constraints	F_{ext}	External force (N)
$\boldsymbol{\mu}$	Lagrange multiplier vector for auxiliary variables bound of MMA	$g(\mathbf{x})$	Inequality constraints function
$\boldsymbol{\xi}$	Lagrange multiplier vector for lower bound inequality of MMA	$h(\mathbf{x})$	Equality constraints function
\mathbf{c}_b	local basis vector	h^{load}	Load height (m)
\mathbf{d}_s	Search direction for slack variables	I	Cross-sectional moment of inertia (kg m ²)
		K	Stiffness matrix
		k	Effective length coefficient
		L	Length (m)

L_j	Lower bounds for MMA	r_{leg}	Jackup leg radius (m)
M	Total weight of the structure	S_b	Block matrix in stiffness matrix
M_o	Overturning moment (N m)	T_b	Transformation matrix
R	Outer radius of annulus	T_k	Trust region at iteration k
r	Inner radius of annulus	U_j	Upper bounds for MMA
r_b	Bar radius (metre)		

Abbreviations

ACE	Automatic Crane Engineer.
BFGS	Broyden–Fletcher–Goldfarb–Shanno.
CONLIN	Convex Linearization Method.
IPM	Interior-Point Method.
KKT	Karush-Kuhn-Tucker.
LICQ	Linear independent constraint qualification.
MMA	Method of Moving Asymptotes.
SQP	Sequential Quadratic Programming.

1

Introduction

The demand for wind turbines with a large efficiency in energy generation is increasing. Larger wind turbines are more efficient, since the power of the wind turbines is directly related to the size of the area that the blades cover. Next to that, the wind flows more steadily higher up in the atmosphere. That is why at the moment, offshore wind turbines are being installed with heights up to 230 meters, and the expectation is that even higher ones will be built in the near future [5].



Figure 1.1: A luffing boom crane on a jackup ship.

To install such a wind turbine, large cranes that can lift as high as the wind turbine's height have to be used. The cranes are transported on jackup ships. These are ships that can lift themselves out of the water and stand on their so-called legs, as can be seen in Figure 1.1. Using such a larger crane to install the wind turbines with the conventional luffing boom crane design that is used nowadays, would mean that also the jackup ship has to be replaced by a larger one. This is a considerable additional cost, since a vessel of that size would cost around 325 million euros. In comparison, an already existing ship costs around 150 million euros [6].

That is why the company Tetrahedron is working on a new design for large offshore cranes that can lift higher, without the need for bigger jackup ships.

This can be seen in Figures 1.4 and 1.5, where the height of the Tetrahedron crane is compared to that of the traditional luffing boom crane. Since there is no need to buy a new, bigger ship, Tetrahedron cranes will make it cheaper to install larger offshore wind turbines in the nearby future. This can be seen in Figure 1.2.

Next to that, in the contemporary offshore industry, if a customer wants a slightly different crane design, the engineers spend much time on the design process by doing computations for each tailor-made crane. Since all ships and load cases are different, this is still frequently done in the industry. To speed up this design process, Tetrahedron has developed the software program Automatic Crane Engineer (ACE) to compute the forces working on the crane structure and to optimize the crane for specific loads. ACE is still under development and one of the goals is to extend the optimization module of the program.

The crane is considered to be a truss structure. This means that it consists of bars that are connected to each other with pins, also called joints. Other examples in structural engineering that can be modeled as truss structures are bridges and roofs. In the literature on the optimization of truss structures, three

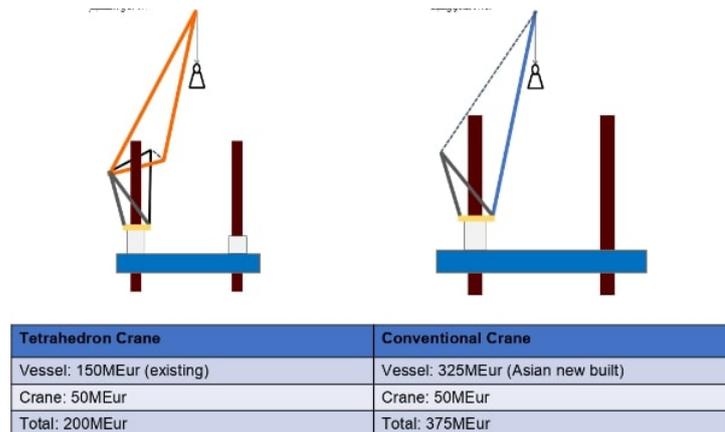


Figure 1.2: Comparison of costs for Tetrahedron crane and conventional luffing boom crane for lifting the 230 meters high wind turbines. By Tetrahedron [6].

types of optimization are considered, separately or in any combination [7–9]. These are size, shape and topology optimization [10]. Figure 1.3 compares the different types of optimization for truss structures. For size optimization, the cross-sectional area of the truss members is optimized. This is considered to be the easiest optimization problem since only one scalar variable per truss member is taken as the design variables, namely the cross-sectional area A . This type of optimization is also already

implemented in ACE.

Shape optimization takes the position of the joints of the structure as design variables. Since size and shape optimization are dependent on each other through certain constraints like buckling, these need to be optimized simultaneously. This combined shape and size optimization will be the topic of interest for this thesis project.

Topology optimization optimizes the connectivity of a so-called ground structure. The ground structure can be either an area consisting of very small elements, or a truss structure with many joints and full connectivity between the joints. Then, these elements are removed to reduce the weight of the object. For Tetrahedron, the topology of the crane is already given and even documented in a patent, so this type of optimization is unnecessary for them.

The literature report that is part of this research [1] describes the background information on mechanical engineering, a mathematical formulation of the problem and different optimization methods. The goal of the research is stated, which is to design a more optimal Tetrahedron crane through a combined shape and size optimization. From this goal, two research questions arise:

- What is the correct formulation of the crane optimization problem?
- What is the best optimization method to solve this crane optimization problem?

To keep the present report self-contained, relevant parts of this literature review report are included. Next to the results of the literature report, the research is extended with more insight into two of the selected methods:

- The Interior-Point Method with a trust-region strategy by Byrd, Hribar and Nocedal [2]

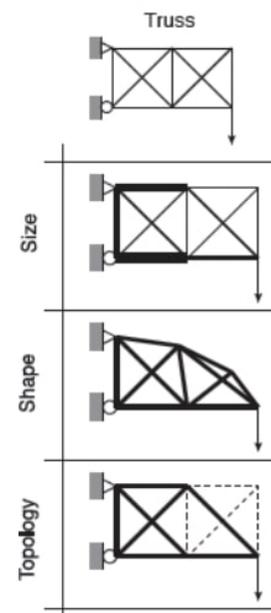


Figure 1.3: Comparison of shape, size, and topology optimization of the truss given above [11]

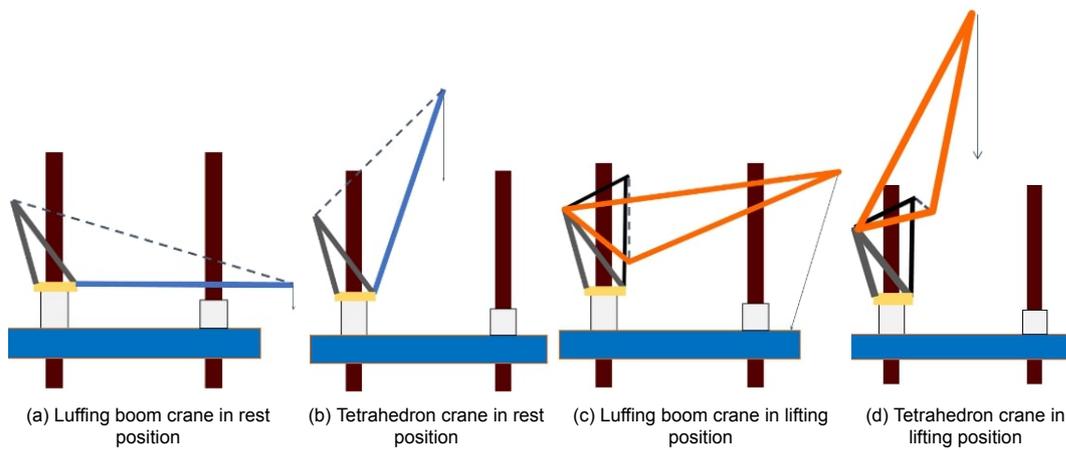


Figure 1.4: Comparison of lifting height and resting position of the luffing boom crane (left) and the Tetrahedron crane (right)

- The Method of Moving Asymptotes by Svanberg [12].

The necessary adaptations to make the methods work smoothly are also explained. The third method that will be tested is the Sequential Least Squares Programming (SLSQP) method by Kraft [4]. This is a variant on Sequential Quadratic Programming and this type of method will shortly be discussed as well, as is done in the literature report. Finally, the results are presented and discussed, and some recommendations on further research are given.



Figure 1.5: Comparison in lifting height of a luffing boom crane (left) and the Tetrahedron crane (right), on the same jackup ship. Clearly, the Tetrahedron crane can install larger wind turbines with the same jackup ship.

2

Crane design and conventions

To analyze the crane mathematically, first some background knowledge about the crane and truss structures is needed. In this chapter, the structure of the crane and its conventions are explained in detail.

2.1. Crane design

Tetrahedron's crane design is found in Figure 2.1 together with its reference axes. The origin of the axes is the center of the slew platform and is indicated by a red point. The slew platform is the black circle in the figure and connects the crane to the ship. The crane can rotate around the z -axis on this slew platform. The elements in black are fixed and can not move. They are constructed around the jackup leg.

The blue-dotted element is called the hoist and controls the movement of the orange elements. This hoist can be shortened or elongated which moves the top of the crane up and down. The crane lifts loads that are attached to the crane in the top. The load is indicated in the figure with blue arrows.

The structure is symmetric in the xz -plane. Figures 2.2 and 2.3 explain the names of the different elements and joints.

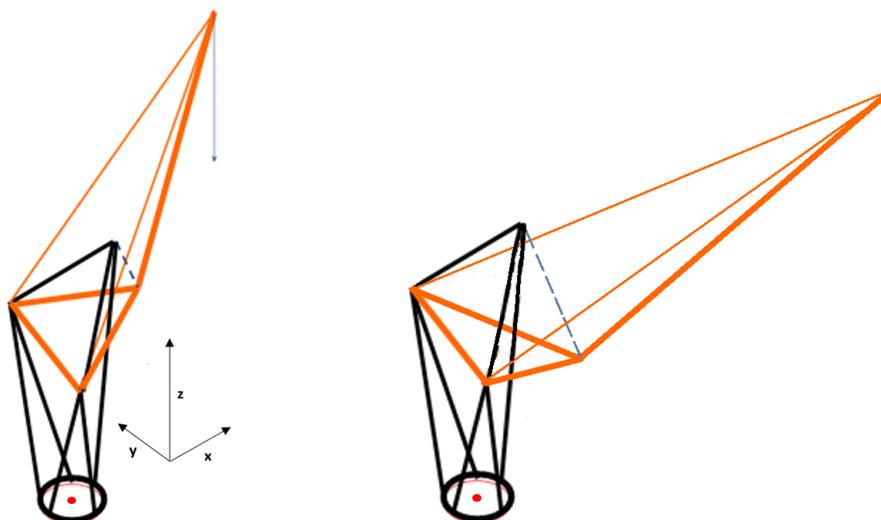


Figure 2.1: Design of the Tetrahedron crane

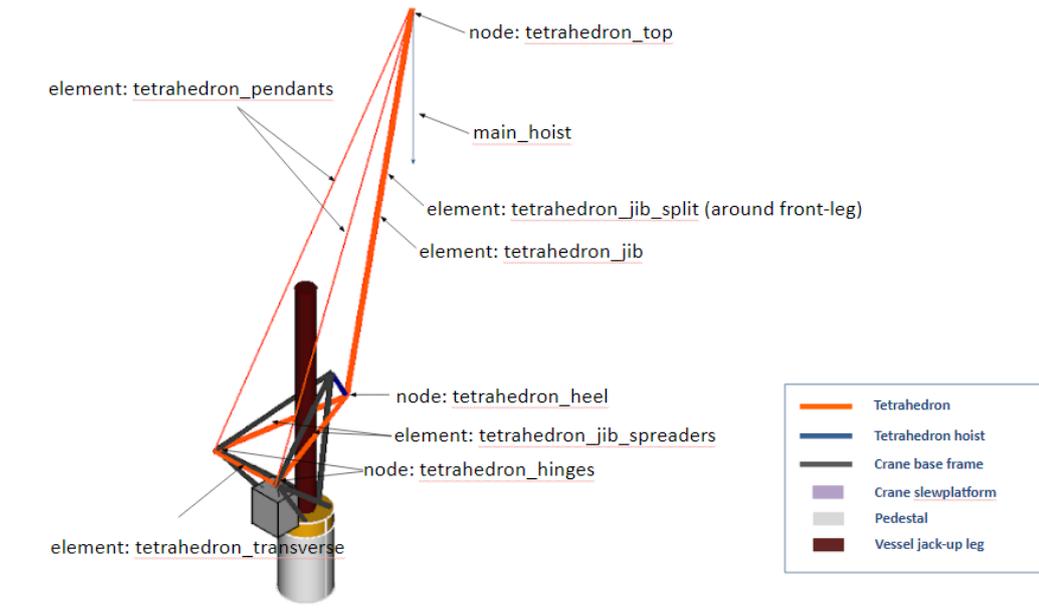


Figure 2.2: Elements and nodes of the crane

2.2. Statics of truss structures

The elements of the truss structure are considered to be bars, which are subjected only to axial load, that is, tensile and compressive forces. This is different from beams, which can also be subjected to torsion or bending moments [13]. The force λ_b in bar b is related to its stress σ_b by the cross-sectional area A_b as

$$\frac{\lambda_b}{A_b} = \sigma_b.$$

If the stresses in the bars become too large, this can lead to structural failure. In the case of compressive forces, the phenomenon of buckling has to be evaded. Buckling means that the compressive forces on the bars are so large that the bar deforms laterally. This happens for forces larger than Euler's critical load F_E .

If a bar is under tension, the stress should not exceed the yield strength σ_{yield} . This indicates the transition from elastic to plastic deformation, such that any force larger than the yield strength will deform the bar permanently, which should by all means be avoided. The yield strength depends on the type of material of the bar. For different types of steel, the yield strength ranges between 250 and 1650 MPa [14].

2.3. Design space

The design is subject to specific features of the crane and mechanical properties. Together these define the amount of freedom in the design variables. The set of all these features is called the design space. The constraints of the optimization problem will follow from this design space.

Next to the mechanical constraints that are discussed in the previous subsection, certain practical constraints emerge due to for example the ship size and the load shape that have to be maintained. This will influence the degrees of freedom of the positions of the joints as well. These practical constraints will be listed here and are elaborated upon in Chapter 3, and they will be supported with schematic drawings of the crane and the ship.

We distinguish three different crane positions that are of interest for the design space. The first is the lowest position when the crane is not being used and the ship can move, the second is the highest position, at which the crane can move objects on the ship deck. The third position that is of interest is the design point of the crane. This is a given height and hoist radius at which the crane should be able

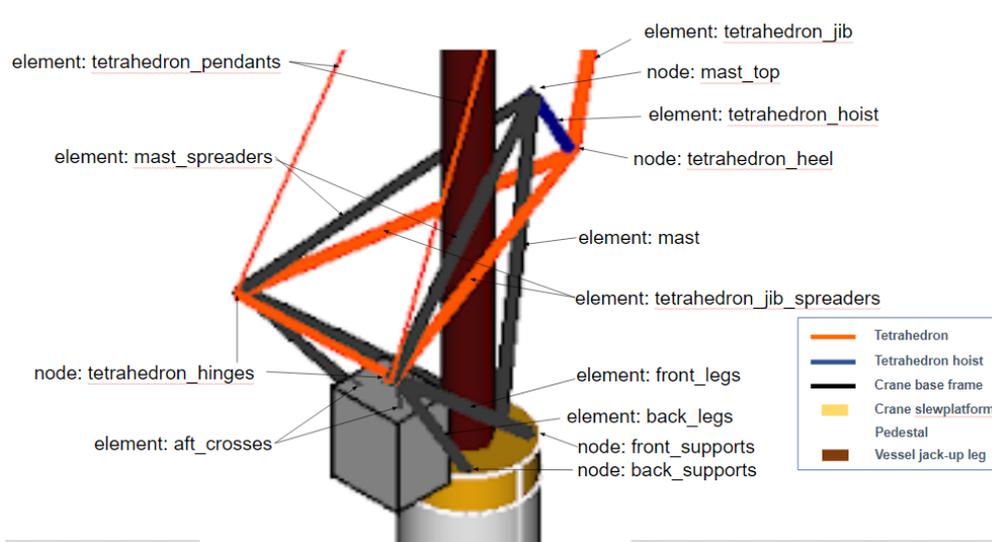


Figure 2.3: Elements and nodes in the base frame

to lift a certain load. The hoist radius is the distance in the x -axis from the Tetrahedron top to the origin of the crane.

Constraints in the lowest crane position

- In its lowest position, the boom of the crane should be in a horizontal position such that transport and maintenance is easy. This can be seen in Figure 2.7.
- The hoist of the crane should not extend such that the jib of the crane will touch the mast, see Figure 3.5.
- The crane will be in its lowest position when the ship is moving. In this case, the structure of the base frame should not have its center of weight too high, or else the ship might not be able to drive safely.

Constraints in the highest crane position

- Here, the crane is defined by a minimum hoist radius at which the crane can lift loads from the deck of the ship.
- There is a minimum clearance between the jib heel and mast top, because the hoist is not able to completely roll up, see Figure 3.5.
- In this highest position, the jib should not touch the mast of the crane as well.

Constraints on the design point

- The hoist radius, height and load that should be lifted are given. For this position and load, the stresses in the bars should not exceed the yield stress or Euler's critical load.
- The load might also be slightly skewed during lifting due to wind which has to be taken into account. A movement under an angle in the xz -plane is called an off-lead, see Figure 2.4c. Similarly, if the load is skewed in the yz -plane, it is called side-lead.
- To control the stresses in the edges under a side lead, the pendants should have a minimal tension on them. This way, the horizontal component of the forces can be handled better, and the crane will not fall over as soon as the maximum side lead is exceeded.
- There should be space between the crane and the hoist that lifts the load, such that the attached object does not collide with the crane. This distance is called the boom clearance. This is also given. See Figure 2.4a.

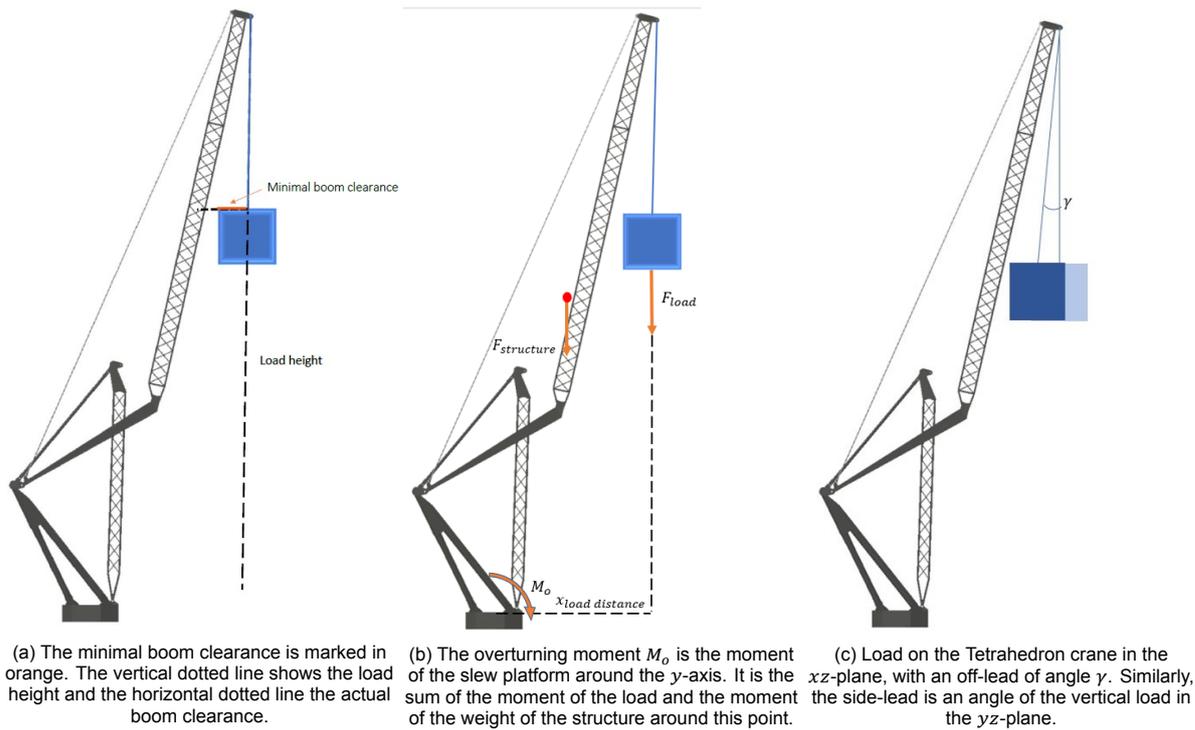


Figure 2.4: Schematic drawings of the xz -plane of the Tetrahedron crane with a load, showing the minimal boom clearance, the overturning moment and the off-lead angle.



Figure 2.5: Conventional crane on a jackup-ship

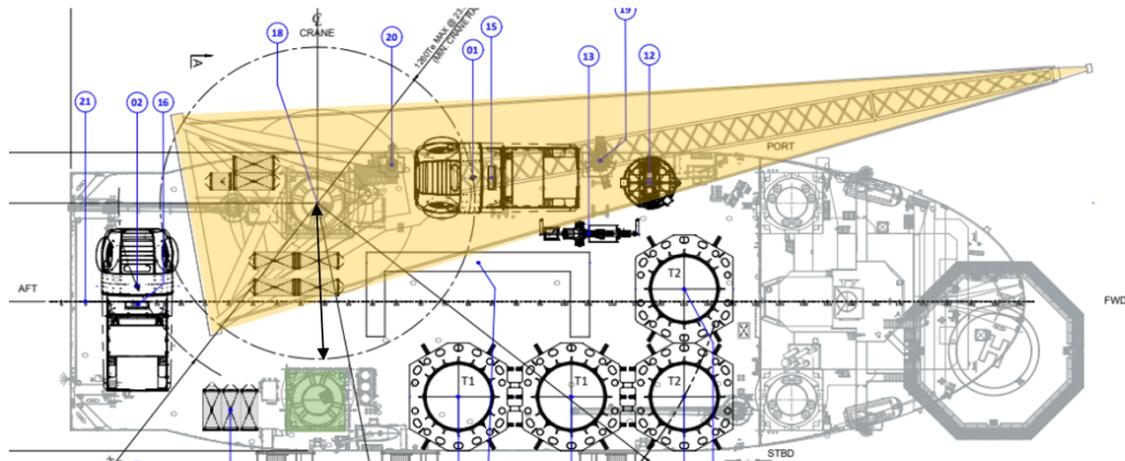


Figure 2.6: Schematic drawing of the xy -plane at the ship deck. The crane is shown in orange, and the other highlighted object in green is the other leg of the jackup ship. The black arrow indicates the maximum safe distance d_{max} .

- The moment at the origin of the crane around the y -axis, should not be too large, or the bearing would break. This moment is called the overturning moment, and its maximum is named the slew bearing moment. See Figure 2.4b.

Constraints in all positions

- The length of the boom of the crane should not be much bigger than the size of the ship it is located on. In Figures 2.1 and 2.5 it can be seen that the lengths of the booms are about as long as the ships for example.
- Since the crane is fixed around one of the legs of the jackup ship, the position of the support elements is fixed on the crane slew platform. This also implies that the bars of the structure should be constructed around the leg without touching it. This can be seen in Figure 3.3.
- The other jackup legs limit the distance of the joints from the origin. The crane will simply not fit on the boat if this limit is exceeded. This is of concern for the hinges of the structure. The distance to the other legs can be seen for a conventional crane in Figure 2.5, and for the Tetrahedron crane in Figure 2.6.
- There is a symmetry in the xz -plane for the nodal positions and the cross-sectional areas of the bars, when the crane is not loaded. This can be seen in figure 2.2. The forces in the bars do not have to be symmetrical however, since the load on the crane can be applied under an angle as well. Therefore the displacement can also be asymmetric.
- The trusses cannot cross each other. This is the case for every truss structure, but here it mainly applies to the Tetrahedron heel and the mast top, see Figure 2.3.

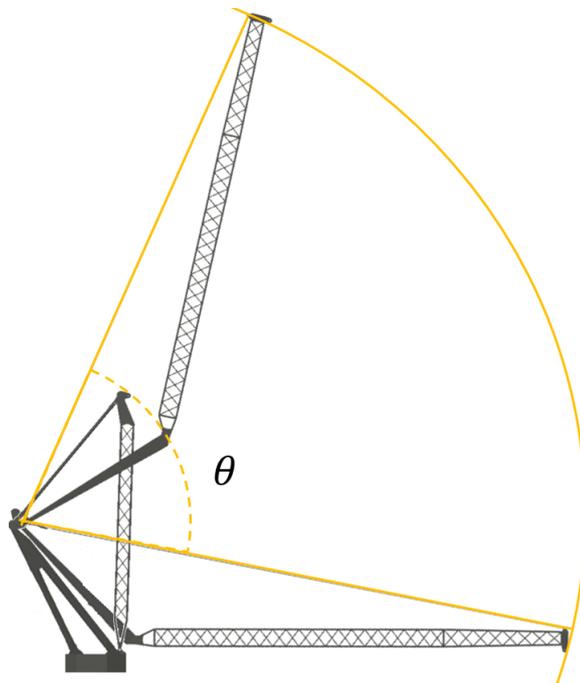


Figure 2.7: Tetrahedron crane as seen in the xz-plane with angle θ from the horizontal position.

3

Mathematical formulation

In this chapter, the optimization problem as explained in Chapter 2 will be translated to a mathematical formulation. The problem that arises will be a constrained continuous optimization problem. The most generic formulation for this type of problem can be described as follows:

$$\begin{aligned} & \text{minimize } f(\mathbf{x}) \\ & \text{subject to } h(\mathbf{x}) = 0 \\ & \quad \quad \quad g(\mathbf{x}) \leq 0, \end{aligned} \tag{3.1}$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $h: \mathbb{R}^n \rightarrow \mathbb{R}^t$, $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$, \mathbf{x} is the vector of design variables, $f(\mathbf{x})$ is the objective function, $h(\mathbf{x})$ is the function of equality constraints, and $g(\mathbf{x})$ is the function of inequality constraints.

The goal of the optimization procedure is to determine the optimal position of the joints of the crane structure and the optimal thickness of the bars. The set of all design variables is therefore defined as

$$\mathbf{x} = \mathcal{N} \cup \{A_b \mid b \in \mathcal{B}\},$$

where \mathcal{N} is the set of all joints \mathbf{n} , A_b is the cross-sectional area of bar b , and \mathcal{B} is the set of bars in the structure. These can be described in terms of the joints, for example $\{\mathbf{n}^{\text{heel}}, \mathbf{n}^{\text{top}}\} \in \mathcal{B}$ defines the bar b^{jib} , see Figure 2.2.

The Tetrahedron crane consists of 10 joints and 16 bars, so without taking the boundary conditions and constraints into account, there would be $3 \cdot 10 + 16 = 46$ design variables. The crane is optimized in the design point, which is a given height and hoist radius at which the crane should be able to lift a given load. Therefore, the Tetrahedron top node is fixed. Next to that, the height of the slew platform is given. This means that the z -coordinate of the supports is also fixed, and not a design variable. Because of symmetry, the 4 nodes in the xz -plane have a fixed y -coordinate at 0. Similarly, the 6 bars that lay outside the xz -plane have a symmetric counterpart and only one of the two has to be taken as a design variable. This reduces the total amount of design variables to 12 variable nodal positions and 10 cross-sectional areas.

3.1. Direct stiffness method

To compute the constraints on the stresses from the forces on the crane, the direct stiffness method is used. This method, which is a special variant of the finite element method [15], computes the displacement \mathbf{u} of the nodes in all directions, corresponding with the load case F . For a truss structure with l directions of displacement, it derives the stiffness equation

$$K\mathbf{u} = F,$$

where $K \in \mathbb{R}^{l,l}$ is the stiffness matrix, $\mathbf{u} \in \mathbb{R}^l$ the displacement vector and $F \in \mathbb{R}^l$ the load vector.

For the crane, the load vector is constructed from three types of loads, namely the weight of the crane itself called the deadweight, the weight of the object that is lifted, called the external force $F_{ext} \in \mathbb{R}$, and the force of the additional masses that are attached to the structure. The additional masses and external weight are given, but the deadweight is dependent on the shape of the structure and is therefore updated during the optimization.

As mentioned in Section 2.3, the external force F_{ext} might be under a small angle γ due to diverse external conditions, for example the wind. This is called the off-lead and/or side-lead angle. In this case, the force is not fully in the z -direction, but has an additional force in the x - and y -direction, as can be seen in Figure 2.4c. The forces on the load node $F_L \in \mathbb{R}^3$ that derive from the external forces are then equal to

$$F_L = \begin{bmatrix} -F_{ext} \cdot \sin(\gamma_{\text{off-lead}}) \\ F_{ext} \cdot \sin(\gamma_{\text{side-lead}}) \\ -F_{ext} \end{bmatrix},$$

where $\gamma_{\text{side-lead}}$ is the side-lead angle, and $\gamma_{\text{off-lead}}$ the off-lead angle.

The stiffness matrix is derived by assembling so-called local stiffness matrices of all bars. The local stiffness matrix in a one-dimensional bar $K_b \in \mathbb{R}^{2,2}$ is dependent on the length L_b of the bar, Young's modulus E_b and the cross-sectional area A_b of the bar:

$$K_b = \frac{E_b A_b}{L_b} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

The length L_b is found as the Euclidean distance of the nodes at the ends of the bar, i.e.

$$L_b = \|\mathbf{n}_{b2} - \mathbf{n}_{b1}\|_2,$$

where \mathbf{n}_{b1} is considered the node at the starting point of bar b and \mathbf{n}_{b2} is the node at the end point of the bar. The choice of which node is the starting or end point, does not matter, as long as it is used consistently during the computation of the stiffness matrix.

The translation from a one-dimensional bar to the three-dimensional space is then done with a transformation matrix $T_b \in \mathbb{R}^{6,2}$ that depends on the local basis vector. The local basis vector $\mathbf{c}_b \in \mathbb{R}^3$ is the unit vector

$$\mathbf{c}_b = \frac{\mathbf{n}_{b2} - \mathbf{n}_{b1}}{L_b}$$

of the bar. The transformation matrix is then given as

$$T_b = \begin{bmatrix} \mathbf{c}_b & 0 \\ 0 & \mathbf{c}_b \end{bmatrix}.$$

The local stiffness matrix for the bar in the 3-dimensional space $K_{lb} \in \mathbb{R}^{6,6}$ is then

$$K_{lb} = T_b K_b T_b^T = \frac{E_b A_b}{L_b} \begin{bmatrix} S_b & -S_b \\ -S_b & S_b \end{bmatrix},$$

where $S_b = \mathbf{c}_b \mathbf{c}_b^T \in \mathbb{R}^{3,3}$ is part of the block matrix K_{lb} .

The global stiffness matrix K is then found by the summation of all contributions K_{lb} of the different trusses.

If K is invertible, we can directly calculate the displacements. However, this is often not possible and a reduction of the matrix is necessary. This is because the nodes that are fixed are still included in the matrix. Since these displacements are always 0, the rows and columns corresponding to these nodes have to be removed. Then, the forces in the bars λ_b can easily be computed with the displacement of the joints of the bar as

$$\lambda_b = \frac{A_b E_b}{L_b} \mathbf{c}_b^T (\mathbf{u}_{b2} - \mathbf{u}_{b1}).$$

where $\mathbf{u}_{b1} \in \mathbb{R}^3$ and $\mathbf{u}_{b2} \in \mathbb{R}^3$ correspond to the displacement at the start and end joint of the truss, respectively.

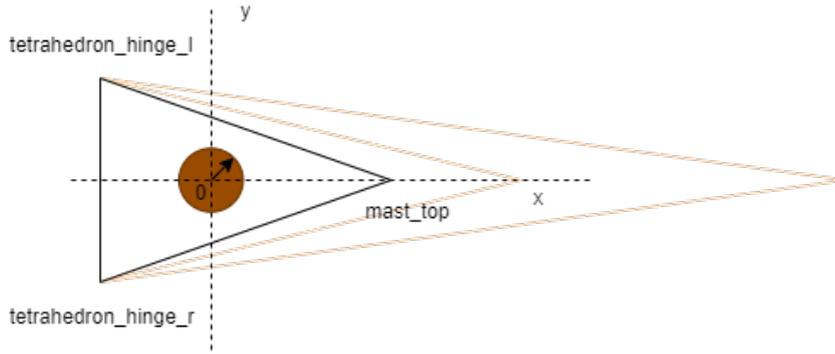


Figure 3.1: Schematic drawing of the xy -plane where elements of the crane are constructed around the jackup leg

3.2. Objective function

The objective is the minimization of the mass of the structure. This will influence the cost of building the structure. The objective function can therefore be denoted as

$$f(\mathbf{x}) = \sum_{b \in \mathcal{B}} A_b L_b(\mathbf{n}) \rho_b, \quad (3.2)$$

where ρ_b is the density of the material, A_b is the cross-section and L_b is the length of bar b , which depends on the nodes \mathbf{n} . The nodes and the cross-sectional area of the bars are the design variables. The values for ρ_b for each bar will be input parameters.

The objective function can be extended with other properties with different weight factors, for example, overturning moment, compliance and price of the materials.

3.3. Constraints

The constraints define the feasible set in which an optimal solution can be found. They are derived from the design space and the mechanical properties that are described in Chapter 2. Instead of rewriting all equations to the form given in Formula (3.1), the equations are denoted such that their meaning is evident. Some of the properties of the crane translate to upper and/or lower bounds of the design variables, while others imply an inequality constraint. First, the inequality constraints will be explained and then the boundary values. Since there are many inequality constraints, they are categorized and explained in different subsections.

Constraints on stresses

For each bar, we have two inequality constraints as a result of the force limits, one for tensile forces and one for compressive forces. We use the convention that a positive force denotes tension and a negative force compression.

As described in Section 2.2, a compressive force should not exceed the critical load. This critical load of the bar, or Euler's critical load, is described by the equation

$$F_E = \frac{\pi^2 EI}{(kL)^2},$$

where L is the bar length, E is Young's modulus and I is the cross-sectional moment of inertia [13]. The effective length factor k is determined by the boundary conditions. For pin ends on both sides of the truss we know $k = 1.0$. This case will be used for truss structures. Young's modulus is a material dependent property that describes the stiffness of a material. It holds that $I = \int_{r \in A} r^2 dA$. Here A is the cross-sectional area. This shows that the critical load depends only on the geometry of the bar and the stiffness of the material [14].

To model this value of I , the cross-sectional shape of the bars are assumed to be annuli with thickness $R - r$, see Figure 3.2. The area moment of inertia for this type of shape is equal to

$$I = \frac{\pi}{4} (R^4 - r^4).$$

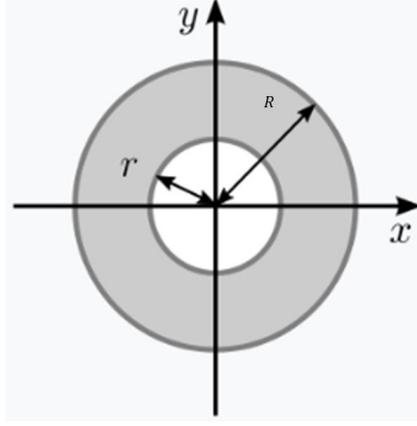


Figure 3.2: Annulus shape of the cross-sectional shape of the bars.

If we set $\alpha r = R$, where $\alpha \in (0, 1)$, we derive the formula

$$I(A) = \frac{A^2}{4\pi} \cdot \frac{1 + \alpha^2}{1 - \alpha^2}, \quad (3.3)$$

where I is dependent on the cross-sectional area A . Reasonable values for α for the crane structures are between 0.95 and 0.99.

Since we derive the stresses in the bars from the displacements, we compute the stress belonging to the Euler force. As described in Section 2.2, these are

$$|\sigma_b| \leq \begin{cases} \sigma_{\text{yield}} & \text{if } \sigma_b \geq 0 \\ \sigma_E = \frac{F_E}{A_b} = \frac{\pi^2 EI}{(kL_b)^2 A_b} & \text{if } \sigma_b < 0. \end{cases} \quad (3.4)$$

Another constraint on the stresses in the pendants is imposed, to reduce the effect that the side lead can have on the structures' forces. This is can be stated as

$$\sigma_{\text{pendants}} \geq \sigma_{\text{pendants}}^{\text{min}}. \quad (3.5)$$

For a maximum side lead of 3 degrees, this minimum pendant stress $\sigma_{\text{pendants}}^{\text{min}}$ in the design point for the Tetrahedron crane is equal to $6.81 \cdot 10^6$ N m.

Constraints on supports

Another inequality constraint comes from the supports on the slew platform. Their position in the xy -plane is bounded between the outer radius of the slew platform and the radius of the leg, see Figure 3.3. It follows that

$$r_{\text{leg}} + r_b \leq \|(n_x, n_y)\|_2 \leq r_{\text{platform}} - r_b \quad \text{for all supports.} \quad (3.6)$$

Here, r_b is the radius of the bar which follows from the cross-sectional area as

$$r_b = \sqrt{\frac{A_b}{\pi \cdot (1 - \alpha^2)}}.$$

Constraint on slew bearing moment

The moment of the crane around the y -axis in the center of the slew bearing should not be too big, or the crane would fall over. This is called the overturning moment M_o . As can be seen in Figure 2.4b, it

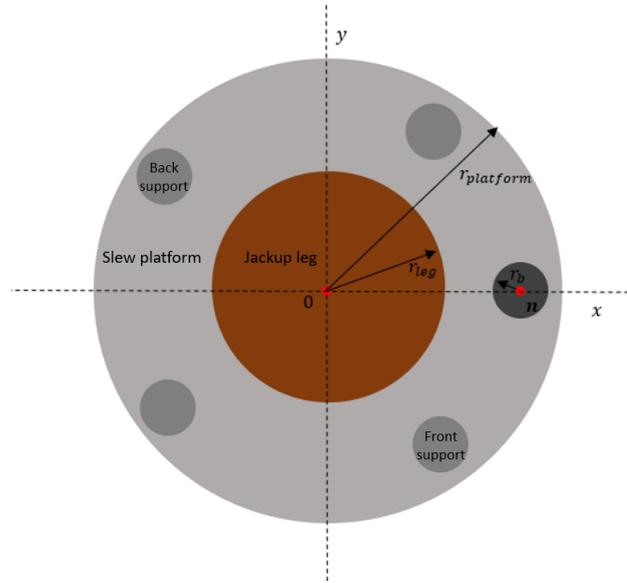


Figure 3.3: Schematic drawing of the xy -plane at the slew platform ($z = 0$). The brown circle represents the leg, the large grey circle the slew platform, and the smaller grey circles represent the bars

is the sum of the moment that derives from the center of mass of the structure, and the moment of the load. It can be computed as

$$M_o = n_x^{\text{top}} \cdot F_{\text{load}} + x_{\text{center of mass}} \cdot M \cdot 9.81,$$

where $x_{\text{center of mass}}$ is the x -component of the center of mass and M is the total weight of the structure, the so-called deadweight.

For the Tetrahedron design, the maximum slew bearing moment M_o^{max} is equal to $6.55 \cdot 10^8$ N m, so the constraint on the slew bearing moment is

$$M_o \leq M_o^{\text{max}} = 6.55 \cdot 10^8 \text{ N m.} \quad (3.7)$$

Constraint on the base frame height

The height of the base frame is limited, because if there is too much weight located high up the ship and the ship rolls due to waves, it might fall over. Therefore, the z -component of the center of gravity of the base frame $z_{\text{center of mass}}^{\text{baseframe}}$ times the weight of this frame is limited by the maximum moment M_z^{max} . The constraint that derives from this is

$$z_{\text{center of mass}}^{\text{baseframe}} \cdot M^{\text{baseframe}} \leq M_z^{\text{max}}. \quad (3.8)$$

The center of gravity of the base frame structure is computed by taking the mean of the edges times its density, cross-sectional area and the gravitational constant.

Constraints on jackup leg distance

The elements should not touch the jackup leg which is in the center of the structure, see Figure 3.1. This means that the distance between the element and the origin in the xy -plane should be greater than r_{leg} . The shortest distance from a line between the points (x_1, y_1) and (x_2, y_2) , to the origin can be described as

$$\frac{|x_2 y_1 - y_2 x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}.$$

This formula is derived from the formula for the area of a triangle $A = \frac{1}{2}bh$, where b is the length of a side and h is the perpendicular height of that side to the top. It follows that the constraint should be

$$\frac{|n_x^2 n_y^1 - n_y^2 n_x^1|}{\sqrt{(n_y^2 - n_y^1)^2 + (n_x^2 - n_x^1)^2}} - r_b \geq r_{\text{leg}} \quad \forall \{\mathbf{n}^1, \mathbf{n}^2\} \in \mathcal{B}. \quad (3.9)$$

Constraint on hinge distance

For the hinges, another constraint on their position is that they should not touch the other legs of the ship, see Figure 2.6. The maximum safe distance to the other leg is denoted by d_{max} which will be given as an input value. This results in the constraint

$$\sqrt{n_x^2 + n_y^2} \leq d_{max}. \quad (3.10)$$

Because of the symmetry constraint on the hinges, it is sufficient to imply this constraint to only one of the hinge joints.

Constraints on jib and mast distances

The clearance between the jib heel and mast top, see Figure 3.5, can be described as

$$\|\mathbf{n}^{heel} - \mathbf{n}^{mast_top}\| \geq d^{hoist}, \quad (3.11)$$

where d^{hoist} is the minimal clearance needed for the hoist.

The boom clearance is the distance between the load and jib in the design point, as shown in Figure 2.4a. Using the ratios of right-angled triangles as shown in Figure 3.4, it is described as

$$d^{boom\ clearance} = \frac{\|n_x^{heel} - n_x^{top}\| \cdot \|n_z^{top} - h^{load}\|}{\|n_z^{heel} - n_z^{top}\|}, \quad (3.12)$$

where h^{load} indicates the load height. Since the boom clearance has to be larger than the minimal boom clearance $d_{min}^{boom\ clearance}$, the constraint on the boom clearance can be written as

$$d^{boom\ clearance} \geq d_{min}^{boom\ clearance}.$$

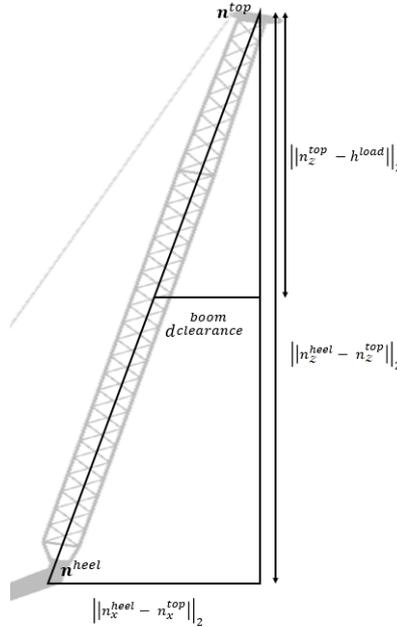


Figure 3.4: Figure of the right-angled triangles used in the calculation of the boom clearance.

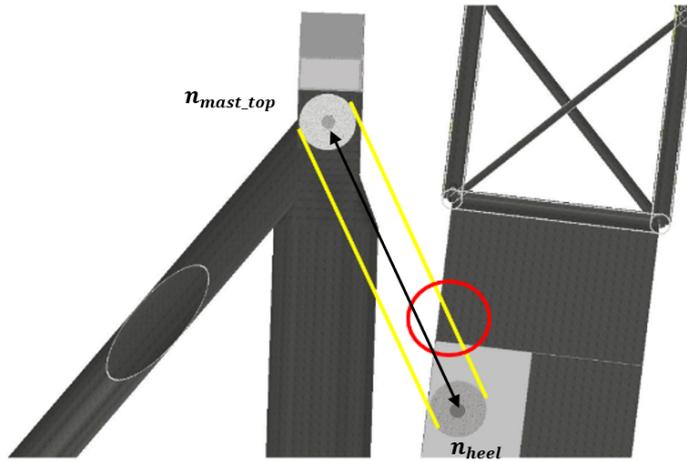


Figure 3.5: Schematic drawing of minimal hoist clearance between the heel and mast top

Constraints on the hoist

First of all, the hoist can at most be in a horizontal position, because after that point, the crane will not be able to lift the jib. This is limited with a constraint that the heel cannot be higher than the mast top. Therefore, at minimum hoist radius, it should hold that

$$n_z^{\text{heel}} - n_z^{\text{mast_top}} \leq 0. \quad (3.13)$$

Next to that, the heel of the Tetrahedron should be in front of the mast bar to physically be able to lift the crane with the hoist. The distance from a point to line is used again but now slightly different because it is necessary to know on which side of the line \mathbf{n}^{heel} is

$$(n_x^{\text{heel}} - n_x^{\text{mast_top}})(n_z^{\text{mast_bottom}} - n_z^{\text{mast_top}}) - (n_z^{\text{heel}} - n_z^{\text{mast_top}})(n_x^{\text{mast_bottom}} - n_x^{\text{mast_top}}) \geq 0. \quad (3.14)$$

How the position of the heel is computed for the minimum and maximum hoist radius is explained in Section 3.4.

3.4. Movement of the crane

Some constraints are not only valid for the design point, but also have to take the movement of the crane into account. This holds for the constraints in Equations (3.11) and (3.14).

The positions of the heel and the top of the crane during the movement can be described with angle θ , as can be seen in Figure 2.7. The minimal hoist radius of the crane is given to determine the movement of the crane. This is the minimal distance from the Tetrahedron top to the origin in the x -direction. Using this radius, the positions will be described by the angle θ around the hinges of the crane. The angle θ for a given hoist radius r is described by

$$\theta(r) = \arccos\left(\frac{r - n_x^{\text{hinge}}}{r_{\text{max}} - n_x^{\text{hinge}}}\right),$$

where the jib of the crane is in horizontal position at the max hoist radius r_{max} .

From this angle, the position of the Tetrahedron top can be determined. In the two-dimensional case, this is equal to

$$\mathbf{n}_{\theta}^{\text{top}} = \mathbf{n}^{\text{hinge}} + \|\mathbf{n}^{\text{hinge}} - \mathbf{n}^{\text{top-design point}}\| \cdot \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}.$$

In the three-dimensional case, a projection to the xz -plane of the coordinates have to be made. Similarly, the position of the heel under angle θ is described by

$$\mathbf{n}_{\theta}^{\text{heel}} = \mathbf{n}^{\text{hinge}} + \|\mathbf{n}^{\text{hinge}} - \mathbf{n}^{\text{heel-design point}}\| \cdot \begin{bmatrix} \cos(\theta - \theta^{\text{heel}}) \\ \sin(\theta - \theta^{\text{heel}}) \end{bmatrix},$$

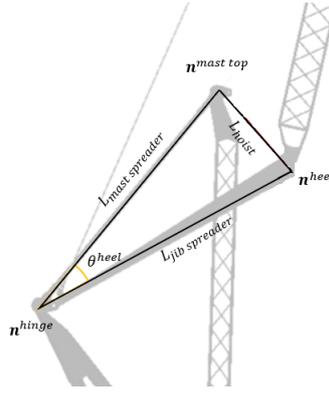


Figure 3.6: Drawing of the angle θ^{heel} and its related nodes.

where θ^{heel} is the angle between the pendants and the jib spreaders in the xz -plane, see Figure 3.6.

For the optimization problem, the minimal and maximal hoist radius are given. For the constraint for the heel to be on the right side (see Equation (3.14)), it is sufficient to check the constraint for the minimal and maximal hoist radius. For the minimal hoist clearance constraint in Equation (3.11), it is sufficient to check only for the minimal hoist radius.

Note that the load that is given only has to be lifted for a part of the range $[0, \theta^{\text{max}}]$, because the crane does not have to carry the same load in all positions.

Total number of constraints

To complete this section, the total amount of constraints for the 3-dimensional Tetrahedron crane case is given.

First of all, the stress constraints (3.4) hold for each bar in the structure. Since only either the compressive or the tensile constraint can hold, these are taken as one constraint on the stresses as

$$\mathbb{1}_{\{\sigma_b \geq 0\}} \cdot (\sigma_b - \sigma_{\text{yield}}) + \mathbb{1}_{\{\sigma_b < 0\}} \cdot (-\sigma_b - \sigma_E) \leq 0. \quad (3.15)$$

This has to hold for all bars, which means there are 16 constraints of type (3.15). The other stress constraint (3.5) only applies to the two pendants, and therefore there are in total 18 constraints on the stresses of the bars.

The constraints on the supports (3.6) of the crane only have to apply to three of the five supports because of symmetry. Because the norm of a support has both an upper and a lower bound, this results in a total of 6 constraints on the supports of the Tetrahedron crane.

The constraint (3.9) only has to be applied to the bars that are close to the leg, namely the jib- and mast spreaders, and the front support legs. Again, the symmetric elements do not have to be taken into account, and therefore there are in total 3 constraints of this type.

The constraint on the hoist (3.14) has to hold at both the maximum and minimum hoist radius, resulting in 2 constraints in total.

The other inequality constraints (3.7), (3.8), (3.10), (3.11), (3.12) and (3.13) refer to only one bar or node, or the structure as a whole, and therefore add only 1 more equality constraint. In total, there are 35 constraints. That is far more than the number of design variables, which is 24, as mentioned in the introduction of this chapter.

3.5. Bounds

Bounds are a special type of constraint that define upper and/or lower limits on the design variables. Most design variables are implicitly limited by the constraints that are explained in Section 3.3. Only a few extra bounds can be set according to physical limitations. First of all, joint $\mathbf{n}^{\text{mast top}}$ should be on

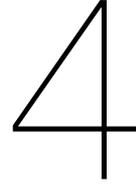
the front side of the leg and the hinges cannot touch the leg, therefore

$$n_x^{\text{mast top}} > r_{\text{leg}}$$

$$n_x^{\text{hinges}} > r_{\text{leg}}$$

$$n_y^{\text{hinges}} > r_{\text{leg}}.$$

The other coordinates and also the cross-sectional areas of the bars are limited by an upper and a lower bound that is set large enough to find reasonable solutions. This is necessary since the Method of moving asymptotes requires upper and lower bounds \underline{x} and \bar{x} , see Section 4.4.



Nonlinear optimization methods

Different optimization algorithms are described in the literature on truss structure optimization. In this research, the focus is on the 'classical' methods for general nonlinear optimization problems as described by Nocedal and Wright [16], and on the class of methods especially developed for structural optimization, for example the Method of Moving Asymptotes (MMA).

In the literature report, a general introduction to the penalty and augmented Lagrangian methods, Sequential Quadratic Programming (SQP) and the Interior-Point Method (IPM) is given, as well as to the Method of Moving Asymptotes and the globally-convergent Method of Moving Asymptotes. A short introduction to SQP methods and especially the SLSQP method by Kraft [4] will be given. Since the preliminary research (see [1]) showed that the Method of Moving Asymptotes and the IPM as implemented by Byrd, Hribar and Nocedal [2] were most promising, these methods will be studied more in-depth here. As an introduction to the Method of Moving Asymptotes, the Convex Linearization Method (CONLIN) method will be explained.

We do not consider the recent trend of optimization methods based on genetic and evolutionary algorithms [8, 17]. These are meta-heuristic algorithms that are based on physical phenomena, such as the Teaching-learning-based algorithm, the Firefly algorithm or the Harmony search algorithm (see [18–20]). However, they do not guarantee convergence to locally optimal solutions [10]. Therefore, they should only be used when gradient-based algorithms are not available or difficult to obtain due to noisiness of the problem or non-smoothness [21].

The methods that will be discussed in this chapter have a similar structure for solving Problem (3.1):

Step 1: Start with initial design variables x , tolerance ε_{tol} , and the method's other parameters. Define optimality measure E .

Step 2: Construct an (easier to solve) subproblem and solve for the approximate solution x^+ .

Step 3: Update $x := x^+$ and parameters accordingly

Step 4: Repeat step 2 and 3 until $E < \varepsilon_{\text{tol}}$

The optimality measure E is a function that checks if the first order optimality conditions are fulfilled. Therefore, we present the necessary conditions for a point $x \in \mathbb{R}^n$ to be a local optimum. From the Karush-Kuhn-Tucker (KKT) theorem, we know that for a constrained problem (3.1), if there is an optimum x^* that holds certain regularity conditions, the KKT conditions have to be satisfied at that point [22]. The Karush-Kuhn-Tucker conditions that have to hold at the optimum (x^*, λ^*) are

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0 \quad (4.1)$$

$$h(x^*) = 0 \quad (4.2)$$

$$g(x^*) \leq 0 \quad (4.3)$$

$$(\lambda_g^*)^T g(x^*) = 0 \quad (4.4)$$

$$\lambda_g^* \geq 0. \quad (4.5)$$

Here, the Lagrangian $\mathcal{L}(x, \lambda_h, \lambda_g)$ is defined as

$$\mathcal{L}(x, \lambda_h, \lambda_g) = f(x) - \lambda_h^T h(x) - \lambda_g^T g(x),$$

where $\lambda_g \in \mathbb{R}^m$ is the vector of the Lagrange multipliers of the inequality constraints, and $\lambda_h \in \mathbb{R}^t$ is the vector of the Lagrange multipliers of the equality constraints. Most numerical methods try to find a point that suffices for these conditions. Note that these are only necessary conditions, and do not imply sufficiency in general.

There are many regularity conditions on x^* that can be used for the KKT-theorem to hold, but the most important and one of the strongest is the Linear independent constraint qualification, which is stated as follows:

Definition 4.1 (Linear independent constraint qualification) *Given the point x^* and the active set $\mathcal{A}(x^*) = \{i \mid g_i(x^*) = 0 \text{ or } h_i(x^*) = 0\}$, then the Linear independent constraint qualification (LICQ) holds if the set of active constraint gradients $\{\nabla g_i(x^*), \nabla h_i(x^*), i \in \mathcal{A}(x^*)\}$ is linearly independent.*

The transpose of the Jacobian matrix of the t equality constraints is

$$A_h(x) = [\nabla h_1(x), \nabla h_2(x), \dots, \nabla h_t(x)]$$

and similarly, $A_g(x)$ is transpose of the Jacobian matrix of the inequality constraints. The assumption of LICQ implies that the matrices $A_g(x)$, $A_h(x)$ have full rank at the solution point x^* . If this is not the case, there are redundant constraints, these can be removed with Gaussian elimination or singular value decomposition techniques.

As said before, the KKT-conditions are used for computing the optimality measure E . Here, the maximum of the infinity-norm of the equality equations of the KKT-conditions at the current point x is often taken, which is

$$E(x) = \max \left(\|\nabla f(x) + A_h(x)\lambda_h + A_g(x)\lambda_g\|_\infty, \|\lambda_g^T g(x)\|_\infty, \|h(x)\|_\infty \right).$$

This will be applied in the Interior-point method that is explained in Section 4.2.

4.1. Sequential Quadratic Programming

For Sequential Quadratic Programming (SQP), the nonlinear problem (3.1) is rewritten to a quadratic subproblem by a quadratic approximation of the Lagrangian and a linear approximation of the constraints. This is because exact solution methods exist for these quadratic optimization problems, as well as good approximate solution methods.

The quadratic subproblem for problem (3.1) is defined as

$$\begin{aligned} \min_x \quad & f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T \nabla_{xx}^2 \mathcal{L}(x^k, \lambda^k) (x - x^k) \\ \text{subject to} \quad & A_h^T (x - x^k) + h(x^k) = 0 \\ & A_g^T (x - x^k) + g(x^k) \leq 0, \end{aligned} \tag{4.6}$$

The choice for these approximations is supported by the KKT-conditions, since they can be seen as an application of the Newton method on the KKT-conditions [16].

In the formulation of Subproblem (4.6), the computation of the Hessian $\nabla_{xx}^2 \mathcal{L}$ is the most expensive. Therefore, instead of computing the Hessian, this can also be approximated by a matrix B_k that is updated each step. An example of such an approximation is the Broyden–Fletcher–Goldfarb–Shanno (BFGS)-method [16].

The Sequential Least Squares Programming (SLSQP) method a variant on the SQP method, which will be used in the two-dimensional experiments in Chapter 5. For this method, the Hessian of the Lagrangian is approximated with a damped BFGS method that guarantees the positive-definiteness of the approximation [4].

To make computations even less expensive, B_k is decomposed as $B_k = L_k D_k L_k^T = L_k D_k^{1/2} (L_k D_k^{1/2})^T$

where L_k is a lower triangular matrix and D_k a diagonal matrix. This is possible because the updated B_{k+1} is always positive definite. Then instead of solving the quadratic subproblem, the linear least squares subproblem

$$\begin{aligned} \min_{\mathbf{d} \in \mathbb{R}^n} & \left\| (D^k)^{\frac{1}{2}} (L^k)^T \mathbf{d} + (D^k)^{-\frac{1}{2}} (L^k)^{-1} \nabla f(\mathbf{x}^k) \right\|_2 \\ \text{subject to} & \quad A_h^T \mathbf{d} + h(\mathbf{x}^k) = 0 \\ & \quad A_g^T \mathbf{d} + g(\mathbf{x}^k) \leq 0 \end{aligned} \quad (4.7)$$

$$(4.8)$$

is solved, where $\mathbf{d} = \mathbf{x} - \mathbf{x}^k$.

The Interior-point method also applies a similar type of approximation as the SQP method, but first reformulates the initial optimization problem. This will be explained in the next section.

4.2. Interior-point methods

The method that will be elaborated upon in this section is the Interior-point method, and in particular the Interior-point method with trust regions as implemented by Byrd, Hribar and Nocedal [2]. Here, a so-called barrier parameter $\mu > 0$ is used to make sure the solution does not come too close to the boundaries of the design space too quickly, and the solution stays in the feasible region. These methods solve the problem

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{s}} & \quad f(\mathbf{x}) - \mu \sum_{i=1}^m \log s_i \\ \text{subject to} & \quad h(\mathbf{x}) = 0 \\ & \quad g(\mathbf{x}) + \mathbf{s} = 0, \end{aligned}$$

where $s > 0$ are slack variables and \log is the natural logarithm. The slack variables are used to transform inequality constraints into equality constraints. By letting μ converge to 0, the solution converges to a feasible optimum. IPM methods are known for good results for problems with a large amount of free variables [16].

The Lagrangian of this problem is

$$\mathcal{L}(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}_h, \boldsymbol{\lambda}_g) = f(\mathbf{x}) - \mu \sum_{i=1}^m \log s_i + h(\mathbf{x})^T \boldsymbol{\lambda}_h + (g(\mathbf{x}) + \mathbf{s})^T \boldsymbol{\lambda}_g. \quad (4.9)$$

The authors of [2] propose to solve Problem (4.9) using a Sequential Quadratic Programming (SQP) method with trust regions. Here, the quadratic approximation of the objective function $\tilde{f}(\mathbf{x}, \mathbf{s}) = f(\mathbf{x}) - \mu \sum_{i=1}^m \log s_i$ around $(\mathbf{x}^k, \mathbf{s}^k)$ is

$$\begin{aligned} \tilde{f}(\mathbf{x}, \mathbf{s}) \approx & f(\mathbf{x}^k) - \mu \sum_{i=1}^m \log s_i^k + \nabla f(\mathbf{x}^k)^T (\mathbf{x} - \mathbf{x}^k) - \mu e^T S_k^{-1} (\mathbf{s} - \mathbf{s}^k) + \\ & \frac{1}{2} \left((\mathbf{x} - \mathbf{x}^k)^T \nabla_{xx}^2 \mathcal{L}(\mathbf{x}^k, \mathbf{s}^k, \boldsymbol{\lambda}_h^k, \boldsymbol{\lambda}_g^k) (\mathbf{x} - \mathbf{x}^k) + (\mathbf{s} - \mathbf{s}^k)^T \Sigma^k (\mathbf{s} - \mathbf{s}^k) \right). \end{aligned} \quad (4.10)$$

Here, Σ^k is a positive definite diagonal matrix that represents either the Hessian of the Lagrangian (4.9) with respect to \mathbf{s} , or an approximation to it. Next to that, e is a vector of ones, and S_k is the diagonal matrix $S_k = \text{diag}(s_1, s_2, \dots, s_n)$.

We look for the search directions \mathbf{d}_x and \mathbf{d}_s , such that the next iterate is found by

$$(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}) = (\mathbf{x}^k, \mathbf{s}^k) + (\mathbf{d}_x, \mathbf{d}_s).$$

Since we look for the next iterate in each step, we can rewrite $(\mathbf{x} - \mathbf{x}^k) = \mathbf{d}_x$ and similarly, $(\mathbf{s} - \mathbf{s}^k) = \mathbf{d}_s$. To improve readability, the arguments of the Lagrangian are also left out hereafter. This leads to the expression

$$\tilde{f}(\mathbf{x}, \mathbf{s}) \approx f(\mathbf{x}^k) - \mu \sum_{i=1}^m \log s_i^k + \nabla f(\mathbf{x}^k)^T \mathbf{d}_x - \mu e^T S_k^{-1} \mathbf{d}_s + \frac{1}{2} (\mathbf{d}_x^T \nabla_{xx}^2 \mathcal{L} \mathbf{d}_x + \mathbf{d}_s^T \Sigma^k \mathbf{d}_s). \quad (4.11)$$

Because we minimize $\tilde{f}(\mathbf{x}, \mathbf{s})$, the first two terms of (4.11) can be dropped since they are independent of \mathbf{d}_x and \mathbf{d}_s . Then, the objective function of the quadratic subproblem is equal to

$$\min_{\mathbf{d}_x, \mathbf{d}_s} \nabla f(\mathbf{x}^k)^T \mathbf{d}_x + \frac{1}{2} \mathbf{d}_x^T \nabla_{xx}^2 \mathcal{L} \mathbf{d}_x - \mu \mathbf{e}^T S_k^{-1} \mathbf{d}_s + \frac{1}{2} \mathbf{d}_s^T \Sigma^k \mathbf{d}_s. \quad (4.12)$$

The first-order Taylor approximation of the constraints is equal to

$$h(\mathbf{x}) \approx A_h(\mathbf{x}^k)^T \mathbf{d}_x + h(\mathbf{x}^k) \quad (4.13)$$

$$g(\mathbf{x}) \approx A_g(\mathbf{x}^k)^T \mathbf{d}_x + \mathbf{d}_s + g(\mathbf{x}^k) + \mathbf{s}^k. \quad (4.14)$$

A distinction can be made between IPM methods that use a line search approach and methods that use a trust-region approach. The method as described by Byrd, Hribar and Nocedal is a trust-region IPM method, which adds another constraint to the problem, namely

$$(\mathbf{d}_x, \mathbf{d}_s) \in T_k. \quad (4.15)$$

where T_k is the trust region. The trust region guarantees that the subproblem always has a finite solution [2].

To determine whether a step heads in the right direction, a merit function ϕ is used. This often checks the two goals of the optimization step: if the objective function value is decreased and if the violation of the constraints is decreased. Here, it is

$$\phi(\mathbf{x}, \mathbf{s}; \nu) = f(\mathbf{x}) - \mu \sum_{i=1}^m \log s_i + \nu \left\| \begin{bmatrix} h(\mathbf{x}) \\ g(\mathbf{x}) + \mathbf{s} \end{bmatrix} \right\|_2. \quad (4.16)$$

Combining (4.12), (4.13), (4.15) and (4.16), the following steps are taken at each iteration:

Step 1: Initiate parameters $\mu > 0$, $\varepsilon_\mu > 0$, $\mathbf{x}_k, \mathbf{s}_k > 0$, $k = 1$, set trust region T_k , compute λ_h, λ_g .

Step 2: Obtain $\mathbf{d}_x, \mathbf{d}_s$ by solving the SQP problem

$$\begin{aligned} \min_{\mathbf{d}_x, \mathbf{d}_s} \nabla f(\mathbf{x}^k)^T \mathbf{d}_x + \frac{1}{2} \mathbf{d}_x^T \nabla_{xx}^2 \mathcal{L} \mathbf{d}_x - \mu \mathbf{e}^T S_k^{-1} \mathbf{d}_s + \frac{1}{2} \mathbf{d}_s^T \Sigma^k \mathbf{d}_s \\ \text{subject to} \quad A_h(\mathbf{x}^k)^T \mathbf{d}_x + h(\mathbf{x}^k) = \mathbf{r}_h \\ A_g(\mathbf{x}^k)^T \mathbf{d}_x + \mathbf{d}_s + g(\mathbf{x}^k) + \mathbf{s}^k = \mathbf{r}_g \\ (\mathbf{d}_x, \mathbf{d}_s) \in T_k \end{aligned} \quad (4.17)$$

Step 3: If $\phi(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}, \nu)$ is sufficiently smaller than $\phi(\mathbf{x}^k, \mathbf{s}^k, \nu)$, update $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}_x$, $\mathbf{s}^{k+1} = \mathbf{s}^k + \mathbf{d}_s$ and compute new λ_h, λ_g , possibly increase trust region.

Else, update $\mathbf{x}^{k+1} = \mathbf{x}^k$, $\mathbf{s}^{k+1} = \mathbf{s}^k$ and decrease trust region.

Step 4: Set $k = k + 1$.

Step 5: Repeat step 2-4 until $E(\mathbf{x}^k, \mathbf{s}^k; \mu) \leq \varepsilon_\mu$, where

$$E(\mathbf{x}^k, \mathbf{s}^k; \mu) = \max \left(\|\nabla f(\mathbf{x}) + A_h^T(\mathbf{x}) \lambda_h + A_g^T(\mathbf{x}) \lambda_g\|_\infty, \|S \lambda_g - \mu \mathbf{e}\|_\infty, \|h(\mathbf{x})\|_\infty, \|g(\mathbf{x}) + \mathbf{s}\|_\infty \right)$$

The norms in the computation of E are directly related to the KKT conditions, which in this case are equal to

$$\nabla_x f(\mathbf{x}) + A_h^T \lambda_h + A_g^T \lambda_g = 0 \quad (4.18)$$

$$-\mu S^{-1} \mathbf{e} + \lambda_g = 0 \quad (4.19)$$

$$h(\mathbf{x}) = 0 \quad (4.20)$$

$$g(\mathbf{x}) + \mathbf{s} = 0 \quad (4.21)$$

$$\lambda_g, \mathbf{s} \geq 0. \quad (4.22)$$

Equations (4.18)-(4.19) follow from the fact that the derivative of the Lagrangian is zero at an optimal point, see Equation (4.1). System (4.18) - (4.22) is called the primal system.

Most of the work of the algorithm is done in the second step when solving the subproblem (4.17). If the solution of (4.17) lies in the trust region, the residuals r_h, r_g would be 0 and the Hessian $\nabla_{xx}\mathcal{L}$ positive definite, in which case the problem can be solved with Newton's method. This method finds a root of a function $F(x)$ by solving a system of the form

$$\nabla F(x^k)(x^{k+1} - x^k) = -F(x^k)$$

In general, this method has quadratic convergence. For the optimization, we take the KKT-system (4.18)-(4.22) as the function F , which finds a root for $d_x, d_s, \lambda_g, \lambda_h$. This results into the system

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L} & 0 & A_h(x_k) & A_g(x_k) \\ 0 & \Sigma_k & 0 & I \\ A_h^T(x) & 0 & 0 & 0 \\ A_g^T(x_k) & I & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_s \\ \lambda_h^+ \\ \lambda_g^+ \end{bmatrix} = - \begin{bmatrix} \nabla_x f(x) \\ -\mu S^{-1}e \\ h(x) \\ g(x) + s \end{bmatrix}, \quad (4.23)$$

where Σ_k is the Hessian with respect to s . For this primal system, Σ_k is equal to the diagonal matrix containing the second derivatives of the Lagrangian with respect to s , which is

$$\Sigma_k = \frac{\partial^2 \mathcal{L}}{\partial s^2} = \frac{\partial}{\partial s} (-\mu S^{-1}e + \lambda_g) = \mu S_k^{-2}.$$

Since we only take the equality equations of the KKT conditions into account with the Newton method, the nonnegativity of Equation (4.22) cannot be guaranteed. If we rewrite condition (4.19) to

$$S\lambda_g - \mu e = 0 \quad (4.24)$$

by multiplying with S , this forces $s_i \cdot \lambda_{g,i}$ to take positive values. Equations (4.18), (4.20)-(4.22) together with (4.24) form the so-called primal-dual system. Even though Equations (4.19) and (4.24) are mathematically equivalent, the latter has proven to perform better computationally [2]. This is because for (4.24), the derivative with respect to s is λ_g , which remains bounded as $s_i \rightarrow 0$.

For the primal-dual system with Equation (4.24) instead of (4.19), the second line of the Newton's system would change into

$$\Lambda_g d_s = -S\lambda_g + \mu e, \quad (4.25)$$

which can be rewritten as

$$S^{-1}\Lambda_g d_s + \lambda_g = \mu S^{-1}e. \quad (4.26)$$

This shows that in the primal-dual case, the system (4.23) can be used, but now $\Sigma_k = S^{-1}\Lambda_g$. This performs better when $s \rightarrow 0$ than the Σ_k for the primal system.

In the selected algorithm, the primal-dual approach is used, since it is likely to perform better. From the computational research, it tends that this method is less likely to violate the constraint of nonnegativity for the slack variables than the primal method, while it still moves into the direction of the solution [2].

However, we cannot assume that the solution of the subproblem lies in the trust region T_k with Newton's method as explained above. This is why residuals r_h, r_g are defined. The problem is divided into two steps, the so-called vertical and horizontal step, that make sure that a solution is found in the trust region. This approach is closely related to the approach taken for solving the system that derives from the MMA method, and will be explained in Section 4.5.

4.3. Convex linearization methods

As an introduction to the Method of Moving asymptotes, the principle of Convex linearization methods will be explained. These methods are based on the usage of a different type of quasi-linear approximations,

which assures convexity of the approximation. The method is developed for optimization problems with only inequality constraints, described as

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad (\mathbf{x} \in \mathbb{R}^n) \quad (4.27)$$

$$\text{subject to } g(\mathbf{x}) \leq 0, \quad (4.28)$$

$$\underline{x}_j \leq x_j \leq \bar{x}_j \quad j = 1, \dots, n \quad (4.29)$$

The approximation is not actually linear, because it also contains reciprocals for some of the variables x_j , to construct a convex approximation. This first-order convex approximation for the objective function (4.27) is equal to

$$\tilde{f}^k(\mathbf{x}) = f(\mathbf{x}^k) + \sum_{+} \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \cdot (x_j - x_j^k) - \sum_{-} (x_j^k)^2 \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \left(\frac{1}{x_j} - \frac{1}{x_j^k} \right) \quad (4.30)$$

where the symbols \sum_{+} , \sum_{-} mean to sum over the terms for which $\frac{\partial f(\mathbf{x}^k)}{\partial x_j}$ is positive or negative, respectively. It is clear that at the point \mathbf{x}^k , we get $f^k(\mathbf{x}^k) = f(\mathbf{x}^k)$. The derivative of (4.30) is

$$\frac{\partial \tilde{f}^k(\mathbf{x})}{\partial \mathbf{x}} = \sum_{+} \frac{\partial f(\mathbf{x})}{\partial x_j} - \sum_{-} (x_j^k)^2 \frac{\partial f(\mathbf{x})}{\partial x_j} \cdot \frac{-1}{x_j^2}$$

which is equal to $\frac{\partial f(\mathbf{x}^k)}{\partial \mathbf{x}}$ for $\mathbf{x} = \mathbf{x}^k$. This shows that $\tilde{f}^k(\mathbf{x})$ is indeed a first order approximation. The function is also convex, since for all $t \in (0, 1)$, it holds that

$$\begin{aligned} \tilde{f}^k(t\mathbf{y} + (1-t)\mathbf{z}) &= f(\mathbf{x}^k) + \sum_{+} \frac{\partial f(\mathbf{x}^k)}{\partial x_j} (ty_j + (1-t)z_j - x_j^k) - \sum_{-} (x_j^k)^2 \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \left(\frac{1}{ty_j + (1-t)z_j} - \frac{1}{x_j^k} \right) \\ &= t \cdot f(\mathbf{x}^k) + (1-t) \cdot f(\mathbf{x}^k) + \sum_{+} \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \cdot (ty_j - tx_j^k + (1-t)z_j - (1-t)x_j^k) - \\ &\quad \sum_{-} (x_j^k)^2 \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \left(\frac{1}{t(y_j - z_j) + z_j} - \frac{1}{x_j^k} \right) \\ &\leq t \left[f(\mathbf{x}^k) + \sum_{+} \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \cdot (y_j - x_j^k) - \sum_{-} (x_j^k)^2 \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \left(\frac{1}{y_j} - \frac{1}{x_j^k} \right) \right] + \\ &\quad (1-t) \left[f(\mathbf{x}^k) + \sum_{+} \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \cdot (z_j - x_j^k) - \sum_{-} (x_j^k)^2 \frac{\partial f(\mathbf{x}^k)}{\partial x_j} \left(\frac{1}{z_j} - \frac{1}{x_j^k} \right) \right] \\ &= t\tilde{f}^k(\mathbf{y}) + (1-t)\tilde{f}^k(\mathbf{z}), \quad \forall \mathbf{y}, \mathbf{z} \in \mathbb{R}^n. \end{aligned}$$

Here, to show that $\frac{1}{t(y_j - z_j) + z_j} \leq \frac{t}{y_j} + \frac{1-t}{z_j} = \frac{t(z_j - y_j) + y_j}{y_j z_j}$, the fractions are cross-multiplied as

$$\begin{aligned} y_j z_j &\leq -t^2(y_j - z_j)^2 + ty_j(y_j - z_j) + tz_j(z_j - y_j) + y_j z_j \\ &= (t - t^2)(y_j - z_j)^2 + y_j z_j \\ &= t(1-t)(y_j - z_j)^2 + y_j z_j \end{aligned}$$

where both $t(1-t) \geq 0$ and $(y_j - z_j)^2 \geq 0$. Similar to (4.30), for the constraints, the convex approximation is equal to

$$\tilde{g}_i^k(\mathbf{x}) = g_i(\mathbf{x}^k) + \sum_{+} \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} \cdot (x_j - x_j^k) - \sum_{-} (x_j^k)^2 \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} \left(\frac{1}{x_j} - \frac{1}{x_j^k} \right). \quad (4.31)$$

This method is developed by Fleury and Braibant [23] and is proven to be the most conservative approximation amongst all approximations that combine direct and reciprocal variables [24].

Based on this type of quasi-linear approximation, Fleury developed CONLIN especially for structural optimization problems, which solves the subproblems in the dual space. The dimension of this dual space is relatively low for problems with many design variables and less constraints. That makes the method very useful for these kind of optimization problems. Another advantage is that the intermediate solutions of this method that are found by solving the subproblems iteratively, are steadily improving feasible designs [25].

Because the Hessian entry is 0 for the so-called 'fixed' variables, it is difficult to solve the problem with Newton-like methods, since the Hessian might be singular. The convergence of the CONLIN method is proven for concave initial objective functions and constraints, but it is also shown that a generalization of this convergence to other problems is impossible. A more general approach like the Method of Moving Asymptotes (MMA) can avoid the necessary assumption of concave functions.

4.4. Method of Moving Asymptotes

The Method of Moving Asymptotes is developed in 1987 by Krister Svanberg, especially for solving nonlinear structural optimization problems as we are dealing with in this research [12]. The method is a generalization of the CONLIN method and is suited well for problems with a not too large number of constraints [26].

In each iteration, a subproblem is generated by so-called 'moving asymptotes', and then solved. The examples in [12] show that the convergence of the process is sped up with respect to the CONLIN method and the overall process is stabilized.

Since the method is originally described for only inequality constraints, the same problem as (4.27)-(4.29) is treated in this section. However, to ensure solvability of the problem, the auxiliary variables $\mathbf{y} \in \mathbb{R}^m$ are added to the problem as

$$\min_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}) + \sum_{i=1}^m (c_i y_i + \frac{1}{2} y_i^2) \quad (4.32)$$

$$\text{subject to } g(\mathbf{x}) - \mathbf{y} \leq 0 \quad (4.33)$$

$$\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}, \mathbf{y} \geq 0, \quad (4.34)$$

with c_i a large number, to ensure that in the optimum, $\mathbf{y} = 0$. If there is no feasible optimum for the original problem, the solution that is given by the method has some $y_i > 0$.

The steps that the algorithm takes are:

Step 1: Initiate $\mathbf{x}, \mathbf{y}, \underline{\mathbf{x}}, \bar{\mathbf{x}}, \mathbf{L}^k, \mathbf{U}^k, \boldsymbol{\alpha}^k, \boldsymbol{\beta}^k$. Here, $\boldsymbol{\alpha}^k$ and $\boldsymbol{\beta}^k$ are the move limits and should be chosen such that

$$L_j^k < \alpha_j^k < x_j < \beta_j^k < U_j^k \quad \forall j = 1, \dots, n$$

Step 2: Solve the subproblem

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{y}} \tilde{f}^k(\mathbf{x}) + \sum_{i \in J} (c_i y_i + \frac{1}{2} y_i^2) \\ \text{subject to } \tilde{g}^k(\mathbf{x}) - \mathbf{y} \leq -\mathbf{r}^k \\ \boldsymbol{\alpha}^k \leq \mathbf{x} \leq \boldsymbol{\beta}^k \\ \mathbf{y} \geq 0 \end{aligned} \quad (4.35)$$

where \tilde{f}^k and \tilde{g}^k are approximations of f and g for iteration k .

Step 3: Update \mathbf{x}, \mathbf{y} and repeat Step 1-2 until convergence.

The name 'moving asymptotes' comes from the chosen bounds $L_j^k < x_j < U_j^k$, where both L_j and U_j can vary during the iterations. The choice of these asymptotes is very important for the behavior of the method. Different choices for L_j^k and U_j^k are made in the literature [12, 27]. For simplicity of the

notation, the superscript k for L_j , U_j , \mathbf{r} , \tilde{f} and \tilde{g} will be left out hereafter. In each iteration k , the values for $\tilde{g}(\mathbf{x})$ are approximated by

$$\tilde{g}_i(\mathbf{x}) = \sum_{j=1}^n \left(\frac{p_{ij}}{U_j - x_j} + \frac{q_{ij}}{x_j - L_j} \right)$$

where

$$p_{ij} = \begin{cases} (U_j - x_j^k)^2 \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} & \text{if } \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} > 0 \\ 0 & \text{if } \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} \leq 0 \end{cases}$$

$$q_{ij} = \begin{cases} 0 & \text{if } \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} \geq 0 \\ -(x_j^k - L_j)^2 \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} & \text{if } \frac{\partial g_i(\mathbf{x}^k)}{\partial x_j} < 0. \end{cases}$$

As can be seen, the value of the derivative of g_i at \mathbf{x}^k determines whether p_{ij} or q_{ij} is equal to zero. This is similar to CONLIN, since the approximation function $\tilde{g}_i^k(\mathbf{x})$ can also be rewritten to

$$\tilde{g}_i(\mathbf{x}) = \sum_{+} \frac{p_{ij}}{U_j - x_j} + \sum_{-} \frac{q_{ij}}{x_j - L_j}.$$

A similar approximation $\tilde{f}^k(\mathbf{x})$ is computed for the objective function $f(\mathbf{x})$. The residual is defined as

$$r_i = g_i(\mathbf{x}^k) - \sum_{j=1}^m \left(\frac{p_{ij}}{U_j - x_j^k} + \frac{q_{ij}}{x_j^k - L_j} \right).$$

The advantage of this quasi-linear approximation $\tilde{g}^k + \mathbf{r}^k$ is that it is convex. It is a first order approximation, since

$$\tilde{g}^k(\mathbf{x}^k) = g(\mathbf{x}^k) \quad \text{and} \quad \frac{\partial \tilde{g}^k}{\partial x_j} = \frac{\partial g}{\partial x_j} \quad \text{at } \mathbf{x} = \mathbf{x}^k.$$

Moreover, since the second derivatives of $\tilde{g}_i^k + \mathbf{r}^k$ are given by

$$\frac{\partial^2 g_i}{\partial x_j^2} = \frac{2p_{ij}}{(U_j - x_j)^3} + \frac{2q_{ij}}{(x_j - L_j)^3}$$

$$\frac{\partial^2 g_i}{\partial x_l \partial x_j} = 0 \quad \text{if } j \neq l$$

and p_{ij} and q_{ij} are both non-negative, the second derivative is positive definite and therefore the approximation is a convex function. Because of this convexity, the KKT-conditions are not only necessary but also sufficient conditions for optimality [16]. This means, that if a point \mathbf{x}^* fulfills the KKT-conditions, it is an optimum. At the point $\mathbf{x} = \mathbf{x}^k$, the second derivatives are equal to

$$\frac{\partial^2 g_i}{\partial x_j^2} = \begin{cases} \frac{2\partial g_i/\partial x_j}{U_j - x_j^k} & \text{if } \frac{\partial g_i}{\partial x_j} > 0 \\ -\frac{2\partial g_i/\partial x_j}{x_j^k - L_j} & \text{if } \frac{\partial g_i}{\partial x_j} < 0. \end{cases}$$

This shows that the second derivatives are larger if L_j and U_j are chosen closer to the approximation x_j^k . Conversely, if L_j and U_j are chosen very far away from x_j^k , then the approximation becomes almost linear.

For numerical stability, the values for p_{ij} and q_{ij} are actually taken as

$$p_{ij} = (U_j - x_j^k)^2 \left(1.001 \left(\frac{\partial g_i}{\partial x_j}(\mathbf{x}^k) \right)^+ + 0.001 \left(\frac{\partial g_i}{\partial x_j}(\mathbf{x}^k) \right)^- + \frac{10^{-5}}{\bar{x}_j - \underline{x}_j} \right)$$

$$q_{ij} = (x_j^k - L_j)^2 \left(0.001 \left(\frac{\partial g_i}{\partial x_j}(\mathbf{x}^k) \right)^+ + 1.001 \left(\frac{\partial g_i}{\partial x_j}(\mathbf{x}^k) \right)^- + \frac{10^{-5}}{\bar{x}_j - \underline{x}_j} \right).$$

Here,

$$\left(\frac{\partial g_i}{\partial x_j}(\mathbf{x}^k)\right)^+ = \max\left\{\frac{\partial g_i}{\partial x_j}(\mathbf{x}^k), 0\right\}, \quad \text{and} \quad \left(\frac{\partial g_i}{\partial x_j}(\mathbf{x}^{(k)})\right)^- = \max\left\{-\frac{\partial g_i}{\partial x_j}(\mathbf{x}^{(k)}), 0\right\}.$$

The values for α_k and β_k are chosen such that the next step is close enough to the previous one, and they are defined as

$$\begin{aligned} \alpha_j^k &= \max\left\{\underline{x}_j, L_j + 0.1(x_j^k - L_j), x_j^k - 0.5(\bar{x}_j - \underline{x}_j)\right\} \\ \beta_j^k &= \min\left\{\bar{x}_j, U_j - 0.1(U_j - x_j^k), x_j^k + 0.5(\bar{x}_j - \underline{x}_j)\right\}. \end{aligned}$$

The last parameters that have to be defined are the upper and lower asymptotes U_j and L_j . They are defined through the former two iterations $k - 1$ and $k - 2$ as

$$\begin{aligned} L_j^k &= x_j^k - \gamma_j^k (x_j^{(k-1)} - L_j^{(k-1)}) \\ U_j^k &= x_j^k + \gamma_j^k (U_j^{(k-1)} - x_j^{(k-1)}) \end{aligned},$$

where

$$\gamma_j^k = \begin{cases} 0.7 & \text{if } (x_j^k - x_j^{(k-1)}) (x_j^{(k-1)} - x_j^{(k-2)}) < 0 \\ 1.2 & \text{if } (x_j^k - x_j^{(k-1)}) (x_j^{(k-1)} - x_j^{(k-2)}) > 0 \\ 1 & \text{if } (x_j^k - x_j^{(k-1)}) (x_j^{(k-1)} - x_j^{(k-2)}) = 0 \end{cases}.$$

It is clear that for $\gamma_j^k > 1$, the interval between the lower and upper asymptotes will become larger compared to $\gamma_j^k = 1$, and that this interval decreases for $0 < \gamma_j^k < 1$. For the iterations $k = 1$, $k = 2$, the parameter γ_j^k is chosen as 0.5.

Now that the subproblem is completely defined, the approach to solving the problem will be explained. First, the Lagrangian of (4.35) is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\mu}) = \tilde{f}(\mathbf{x}) + \sum_{i=1}^m c_i y_i + \frac{1}{2} y_i^2 + \sum_{i=1}^m \lambda_i (\tilde{g}_i(\mathbf{x}) - y_i + r_i) + \sum_{j=1}^n (\xi_j (\alpha_j - x_j) + \eta_j (x_j - \beta_j)) - \sum_{i=1}^m \mu_i y_i,$$

where $\boldsymbol{\lambda}$, $\boldsymbol{\xi}$, $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$ are the Lagrange multipliers corresponding to the different inequality constraints of (4.35). This can be rewritten as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\mu}) = \psi(\mathbf{x}, \boldsymbol{\lambda}) + \sum_{j=1}^n (\xi_j (\alpha_j - x_j) + \eta_j (x_j - \beta_j)) + \sum_{i=1}^m \left(c_i y_i + \frac{1}{2} y_i^2 - \lambda_i y_i + \lambda_i r_i - \mu_i y_i \right)$$

where $\psi(\mathbf{x}, \boldsymbol{\lambda}) = \tilde{f}(\mathbf{x}) + \sum_{i=1}^m \lambda_i (\tilde{g}_i(\mathbf{x}) + r_i)$. With this, we can construct the KKT-conditions of the problem. These are equal to

$$\frac{\partial \psi}{\partial x_j} - \xi_j + \eta_j = 0, \quad j = 1, \dots, n \quad (\partial \mathcal{L} / \partial x_j = 0) \quad (4.36)$$

$$c_i + y_i - \lambda_i - \mu_i = 0, \quad i = 1, \dots, m \quad (\partial \mathcal{L} / \partial y_i = 0) \quad (4.37)$$

$$\tilde{g}_i(\mathbf{x}) - y_i + r_i \leq 0, \quad i = 1, \dots, m \quad (4.38)$$

$$\alpha_j \leq x_j \leq \beta_j, \quad j = 1, \dots, n \quad (4.39)$$

$$y_i \geq 0, \quad i = 1, \dots, m \quad (4.40)$$

$$\lambda_i (\tilde{g}_i(\mathbf{x}) - y_i + r_i) = 0, \quad i = 1, \dots, m \quad (4.41)$$

$$\xi_j (\alpha_j - x_j) = 0, \quad j = 1, \dots, n \quad (4.42)$$

$$\eta_j (x_j - \beta_j) = 0, \quad j = 1, \dots, n \quad (4.43)$$

$$-\mu_i y_i = 0, \quad i = 1, \dots, m \quad (4.44)$$

$$\xi_j \geq 0 \text{ and } \eta_j \geq 0, \quad j = 1, \dots, n \quad (4.45)$$

$$\mu_i \geq 0, \quad i = 1, \dots, m \quad (4.46)$$

$$\lambda_i \geq 0, \quad i = 1, \dots, m \quad (4.47)$$

where Equations (4.36)-(4.37) correspond to Equation (4.1), Equations (4.38)-(4.40) to Equation (4.3), Equations (4.41)-(4.44) to Equation (4.4) and lastly, Equations (4.45)-(4.47) to Equation (4.5) of the general KKT-conditions.

As with the interior-point method, this system is solved with a primal-dual method. Therefore, the next section gives a general introduction to primal-dual solution methods.

4.5. Primal-dual solution methods

Primal-dual methods take the approach to update both the design variables, or primal variables, and the Lagrange multipliers, or dual variables, in each iteration, while satisfying $\lambda \geq 0$ elementwise, strictly. This is done by perturbing the Equation (4.4) of the KKT system with the negative of a small parameter $\varepsilon > 0$. By making ε go to 0, the solution of the initial KKT system is attained. For the general problem (3.1) with slack variables introduced for the inequality constraints, this would result in the perturbed KKT conditions

$$\begin{aligned} \nabla_x \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}_g^*, \boldsymbol{\lambda}_h^*) &= 0 \\ h(\mathbf{x}^*) &= 0 \\ g(\mathbf{x}^*) + \mathbf{s} &= 0 \\ (\boldsymbol{\lambda}_g^*)^T g(\mathbf{x}^*) - \varepsilon \mathbf{e} &= 0 \\ \boldsymbol{\lambda}_g^* &> 0 \\ \mathbf{s} &\geq 0. \end{aligned} \quad (4.48)$$

For the interior-point method, the term $\mu \varepsilon$ in Equation (4.24) can be seen as a perturbation of the initial optimization problem, where $\mu \rightarrow 0$.

Conversely, it also holds that Equations (4.48) are the KKT conditions of the convex problem

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{s}} \quad & f(\mathbf{x}) - \varepsilon \sum_{i=1}^m \log s_i \\ \text{subject to} \quad & g(\mathbf{x}) + \mathbf{s} = 0 \\ & h(\mathbf{x}) = 0 \\ & \mathbf{s} > 0. \end{aligned} \quad (4.49)$$

The difference is only that the condition $(\boldsymbol{\lambda}_g^*)^T g(\mathbf{x}^*) - \varepsilon \mathbf{e}$ does not derive from the inequality constraints, but as the derivative of the Lagrangian to \mathbf{s} .

It is clear that the primal-dual approach of the MMA and the IPM are related, but the application of the approach differs between the two methods. MMA uses a line-search method, and the IPM implements a trust-region approach. They will therefore be discussed separately

Primal-dual approach on the Interior Point method subproblem

The primal-dual approach for the interior-point method decomposes the subproblem in two steps: the vertical steps attempts to satisfy the constraints, and the horizontal step searches for an optimal point in the region. The vertical step will be discussed first.

A problem with an optimization method that applies a trust-region instead of a line-search approach, is that it might not be feasible to satisfy the constraints in the region. Therefore, instead of trying to satisfy them exactly, the distance of the constraints is minimized in the sense of least squares. This leads to the system

$$\begin{aligned} \min_{\mathbf{v}} \quad & \|A_h^T \mathbf{v}_x + h(\mathbf{x})\|_2^2 + \|A_g^T \mathbf{v}_x + \mathbf{v}_s + g(\mathbf{x}) + \mathbf{s}\|_2^2 \\ \text{subject to} \quad & \|(\mathbf{v}_x, S^{-1} \mathbf{v}_s)\|_2 \leq \zeta T_k \\ & \mathbf{v}_s \geq -\tau \mathbf{s}. \end{aligned}$$

Here, \mathbf{v} is the vertical step, ζ and τ are parameters between 0 and 1. The first constraint is derived from the facts that the trust region $(\mathbf{v}_x, \mathbf{v}_s)$ has to be smaller than the trust region radius, and that the slack variables should also not approach zero too quickly. Therefore, \mathbf{v}_s is scaled with S^{-1} that penalizes near the boundary. The second constraint implies that the slack variables should remain positive. This constraint is equal to

$$\mathbf{s} + \mathbf{d}_s \geq (1 - \tau) \mathbf{s}$$

where $\tau \in (0, 1)$, often chosen very close to 1. This problem is solved with the dogleg method [16], which is a cheap computation and finds the best feasible vertical step.

The horizontal step uses the vertical step for the definition of the residuals \mathbf{r}_h and \mathbf{r}_g and this together with Equations (4.17) leads to the feasible quadratic subproblem

$$\begin{aligned} \min \quad & \nabla f^T \mathbf{d}_x - \mu e^T S^{-1} \mathbf{d}_s + \frac{1}{2} (\mathbf{d}_x^T \nabla_{xx}^2 \mathcal{L} \mathbf{d}_x + \mathbf{d}_s^T \Sigma \mathbf{d}_s) \\ \text{subject to} \quad & A_h^T \mathbf{d}_x = A_h^T \mathbf{v}_x \\ & A_g^T \mathbf{d}_x + \mathbf{d}_s = A_g^T \mathbf{v}_x + \mathbf{v}_s \\ & \|(\mathbf{d}_x, S^{-1} \mathbf{d}_s)\|_2 \leq T_k \\ & \mathbf{d}_s \geq -\tau \mathbf{s}. \end{aligned}$$

Say $\mathbf{d} = \mathbf{v} + \mathbf{w}$, where \mathbf{d} is the total search direction, then we still need to compute \mathbf{w} , which is tangent

to the constraints. Defining $\tilde{\mathbf{d}} = \begin{bmatrix} \mathbf{d}_x \\ S^{-1} \mathbf{d}_s \end{bmatrix} = \begin{bmatrix} \mathbf{v}_x \\ \tilde{\mathbf{v}}_s \end{bmatrix} + \begin{bmatrix} \mathbf{w}_x \\ \tilde{\mathbf{w}}_s \end{bmatrix}$ as change of variables for the steps and defining $G = \begin{bmatrix} \nabla_{xx}^2 \mathcal{L} & 0 \\ 0 & S \Sigma S \end{bmatrix}$ leads to the optimization problem

$$\begin{aligned} \min_{\tilde{\mathbf{w}}} \quad & q(\tilde{\mathbf{v}} + \tilde{\mathbf{w}}) \equiv q(\tilde{\mathbf{v}}) + \nabla f^T \mathbf{w}_x - \mu e^T \tilde{\mathbf{w}}_s + (G \tilde{\mathbf{v}})^T \tilde{\mathbf{w}} + \frac{1}{2} (\tilde{\mathbf{w}}^T G \tilde{\mathbf{w}}) \\ \text{subject to} \quad & A_h^T \mathbf{w}_x = 0 \\ & A_g^T \mathbf{w}_x + S \tilde{\mathbf{w}}_s = 0 \\ & \|\tilde{\mathbf{w}}\|_2^2 \leq T_k^2 - \|\tilde{\mathbf{v}}\|_2^2 \\ & \tilde{\mathbf{w}}_s \geq -\tau e - \tilde{\mathbf{v}}_s \end{aligned}$$

To solve this problem, the conjugate gradient method is applied to the objective function, while forcing the iterations to satisfy the constraints [2].

which is preferred if there are more variables than constraints. The system only depending on the primal variables x is

$$(D_x + A^T D_{\lambda y}^{-1} A) \Delta x = -\tilde{\delta}_x - A^T D_{\lambda y}^{-1} \tilde{\delta}_{\lambda y}, \quad (4.58)$$

which is preferred if there are more constraints than variables. Here, $D_{\lambda y} = D_\lambda + D_y^{-1}$ and $\tilde{\delta}_{\lambda y} = \tilde{\delta}_\lambda + D_y^{-1} \tilde{\delta}_y$.

The left hand side matrix of equations (4.57) and (4.58) are the Schur complements of

$$\begin{bmatrix} D_x & A^T \\ A & -D_{\lambda y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \tilde{\delta}_x \\ \tilde{\delta}_{\lambda y} \end{bmatrix}.$$

The inverse matrices of D_x , D_y and D_λ are needed for these eliminations, and therefore it is necessary that D_x , D_y and D_λ are well-conditioned in order for the reduced systems to be numerically accurate. From the substitutions, the values for all $\Delta x, \Delta y, \Delta \lambda, \Delta \xi, \Delta \eta, \Delta \mu, s$ can be updated.

The method of moving asymptotes uses a line-search method, where τ is the step size. Let $\mathbf{w} = (x, y, \lambda, \xi, \eta, \mu, s)$, then the iteration k updates the values as

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \tau^k \cdot \Delta \mathbf{w}^k \quad (4.59)$$

and such that the positivity conditions of (4.39)-(4.40) and (4.45)-(4.47) are not violated. To find a good τ , first choose the largest number $t \geq 1$ such that

$$\begin{aligned} x_j + t \Delta x_j - \alpha_j &\geq 0.01 (x_j - \alpha_j) \quad \forall j \\ \beta_j - (x_j + t \Delta x_j) &\geq 0.01 (\beta_j - x_j) \quad \forall j \\ (\mathbf{y}, \lambda, \xi, \eta, \mu, s) + t \cdot (\Delta \mathbf{y}, \Delta \mathbf{z}, \Delta \lambda, \Delta \xi, \Delta \eta, \Delta \mu, \Delta s) &\geq 0.01 \cdot (\mathbf{y}, \mathbf{z}, \lambda, \xi, \eta, \mu, \zeta, s). \end{aligned}$$

The new point should also be better than the last one, and therefore there should be a decrease in the norms of the left-hand sides of the KKT conditions. We set

$$\tau = \{\max(t, t/2, t/4, t/8, \dots) \mid \|\delta(\mathbf{w} + \tau \cdot \Delta \mathbf{w})\|_2 < \|\delta(\mathbf{w})\|_2\},$$

where $\delta(\mathbf{w})$ is the residual vector defined by the left hand sides of the perturbed KKT conditions (4.36)-(4.39), (4.50) and (4.51)-(4.54).

To summarize this section, the steps for the primal-dual method of the Method of Moving Asymptotes are stated. These are taken every time Subproblem (4.35) is solved.

Step 1: Set initial values for \mathbf{w}^k and ε , $k = 1$.

Step 2: Compute $\Delta \mathbf{w}^k$ by solving the perturbed KKT-system with Newton's method

Step 3: Compute step size τ^k and update $\mathbf{w}^{k+1} = \mathbf{w}^k + \tau^k \Delta \mathbf{w}^k$

Step 4: If $\|\delta(\mathbf{w}^{k+1})\|_\infty < 0.9\varepsilon^k$, update ε as $\varepsilon^{k+1} = 0.1\varepsilon^k$, else, $\varepsilon^{k+1} = \varepsilon^k$

Step 5: Repeat Step 2-4 until $\varepsilon \leq 10^{-1}$ and $\|\delta(\mathbf{w}^{k+1})\|_\infty < 0.9\varepsilon^k$.

4.6. Comparison of methods

The methods above all have their advantages and disadvantages, and the selection of one over the other is largely based on the type of nonlinear optimization problem. The methods will be compared in relation to the crane optimization problem.

The interior-point methods often perform well for large-scale applications. However, they are sensitive to the choice of the initial parameters and the manner in which the barrier parameter μ is updated. If the problem is badly scaled, this might be a problem. There are again many different software packages that differ in their approach, and packages that combine SQP and IPM [2].

Schittkowski (2003, [28]) made a comparison between the MMA and SQP method. Since the IPM method that is discussed in Section 4.2 uses a similar quadratic approximation as with SQP, this

comparison is useful in showing the differences between the quasi-linear and second-order approximation subproblems. In the comparison, the gradients are approximated with a five-point difference formula and the comparison is made for the success rate and the number of function evaluations. The function evaluations are a useful performance parameter since for practical applications, this dominates the computation time. For a standard test set of 306 problems, the SQP method clearly outperforms the MMA method in both success rate and function evaluations. The success rate is 100% compared to 93% for the MMA method. This might be because many test cases involve highly nonlinear objective functions, and since the MMA method only applies a first-order approximation, it performs worse.

For specifically structural optimization problems however, the MMA method performs better. It needs less function evaluations than the SQP method, but still the reliability of the method is lower than the SQP method, which has a higher success rate (84% compared to 73% for a tolerance level of 0.01). For very large problems (more than 10^6 variables), the MMA method is still able to solve them, whereas the SQP method is more suited for small optimization problems [28]. Similar results have been found by Schittkowski in 1994 [29], where the SQP method has the highest robustness compared to all other methods, but it can be a bit slower in computation time of structural optimization compared to MMA. A disadvantage of the MMA method is that there are asymptotes for which the initial and updated values influence the computation time.

A summary of the comparison between the methods that are mentioned here can be found in Table 4.1. The column 'Convergence' indicates how often the method finds an optimal point, and the column 'Scalability' indicates how well the algorithm is suited for large-scale problems. The plus- and minus symbols indicate how well the method performs on that property, in comparison to the other methods that are discussed in this Chapter. The scale ranges from – to ++.

Method	Convergence	Scalability	Simplicity
IPM	+	++	+
SQP	++	-	+
MMA	+/-	++	-

Table 4.1: Comparison of methods on convergence, scalability and simplicity

5

2D experiments

In this chapter, the optimization algorithms' performances are compared for four test cases. They are all structural optimization problems where the size and shape of the structures are the design variables, and the stresses in the elements are constrained. All these experiments are in two dimensions, because this is a good check of the algorithms before moving towards the 3-dimensional final problem. The four test cases are:

- The 10-bar truss problem, see Figure 5.1a
- Michell's arch, see Figure 5.1b
- The 2D luffing boom crane, see Figure 5.1c
- The 2D Tetrahedron crane, see Figure 5.1d.

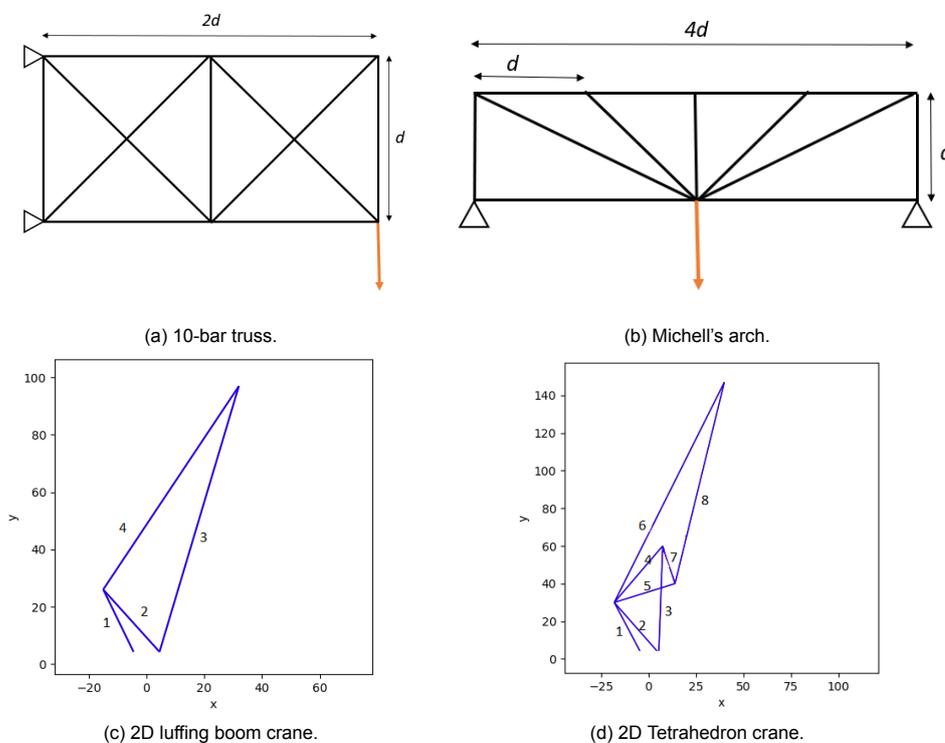


Figure 5.1: Initial structures of the test cases. For the cranes, the elements are numbered. This numbering is used to distinguish the cross-sectional areas, yield strength and stresses of each bar.

The 10-bar truss problem and Michell's arch have been studied in many articles to test algorithms on shape, size and topology optimization (see for example [9, 30, 31]). The weight of the structure is minimized with constraints on the internal forces of the trusses by yield stress and Euler force, and additional upper and lower bounds on the design variables to limit their movement.

The other two test cases are more related to the final problem of this research, and other constraints as explained in Chapter 3 are added to the stress constraints here, such as the constraint on the overturning moment (3.7). All constraints are inequality constraints. The parameters for the cranes are derived from ACE.

5.1. Optimization methods

For the test cases, three methods are selected to compare the results from:

- The least squares SQP method by Kraft [4], named *SLSQP* with the implementation provided in the Python package *scipy.optimize*
- The Interior-Point Method with a trust-region strategy by Byrd, Hribar and Nocedal [2], also with the implementation provided in the python package *scipy.optimize*
- The Method of Moving Asymptotes by Svanberg as described in Section 4.4.

For the Method of Moving Asymptotes, ill-conditioning of the matrix solved in the primal-dual method can be a problem. Therefore, the objective function, constraints and design variables have to be properly scaled. As described in [3], for the parameters that are used in that document, the objective function has to be scaled such that

$$1 \leq f(\mathbf{x}) \leq 100$$

and the constraints should also be scaled such that

$$1 \leq g_i(\mathbf{x}) \leq 100, \quad i = 1, \dots, m$$

and the values of the design variables should be between 0.1 and 100.

If the stress constraints and objective function are not scaled, their values are in the order of 10^8 and 10^5 respectively, which results in an ill-conditioned matrix when solving the problem. Therefore, the objective function is scaled as

$$f(\mathbf{x}) = \sum_{b \in \mathcal{B}} \rho_b A_b L_b \cdot \frac{1}{\rho^{max}}$$

and the stress constraints are scaled as

$$\frac{\sigma_b - \sigma_b^{yield}}{\sigma_{max}^{yield}} \leq 0$$

for tensile stress and as

$$\left(-\frac{\pi^2 E_b I}{(KL_b)^2 A_b} - \sigma_b \right) \cdot \frac{1}{E_{max} \cdot \pi^2} \leq 0$$

for compressive stress.

5.2. Gradient computation

For all methods, the partial derivatives of the objective function and the constraints have to be given as an input by the user, since these are used in the algorithm. If they are not available, they can be approximated numerically by a 2-point finite difference scheme. If the gradients are available, the algorithm might be quicker because it can skip this approximation, and it is of course more accurate. Therefore, the gradient of the objective function (3.2) is derived analytically. The constraints are approximated with the finite difference scheme, since the constraints differ for the various test cases and the derivation of the Jacobian for the stress constraints involves the inverse of the stiffness matrix

in each step, which is computationally expensive. The gradient of the objective function is equal to

$$\nabla \sum_{b \in \mathcal{B}} A_b L_b \rho_b = \nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{n}} \\ \frac{\partial f}{\partial A} \end{bmatrix},$$

where the derivatives of f to the nodal positions is equal to

$$\frac{\partial f}{\partial \mathbf{n}^d} = \sum_{e \in \mathcal{E}} \rho_e A_e \frac{\partial L_e}{\partial \mathbf{n}^d},$$

where $\mathcal{E}(\mathbf{n}^d)$ is the set of edges that are connected to node \mathbf{n}^d , and index $d \in \{1, 2\}$ indicates if node \mathbf{n}^d is marked as the start or end node of the edge, respectively. The derivative of the length of the edge with respect to node \mathbf{n}^d is equal to

$$\frac{\partial L_e}{\partial \mathbf{n}^d} = \begin{cases} \frac{\mathbf{n}_e^1 - \mathbf{n}_e^2}{\|\mathbf{n}_e^2 - \mathbf{n}_e^1\|} & \text{if } d = 1 \\ \frac{\mathbf{n}_e^2 - \mathbf{n}_e^1}{\|\mathbf{n}_e^2 - \mathbf{n}_e^1\|} & \text{if } d = 2. \end{cases}$$

the derivative to the cross-sectional area of a bar b is equal to

$$\frac{\partial f}{\partial A_b} = \rho_b \cdot L_b.$$

Now that the optimization methods are defined, the results of the test cases are presented in Sections 5.3 - 5.6.

5.3. 10-bar truss problem

For the 10-bar truss problem, a vertical load is attached on the bottom-right node and the structure is fixed by the two left-most nodes, as can be seen in Figure 5.1a. The parameters that are chosen for this problem can be found in Table 5.1. This problem has 16 design variables and 10 constraints. The parameter d determines the initial positions for the nodes as can be seen in Figure 5.1a. Choosing $\alpha = 0$ for determining the thickness of the annulus of the cross-sectional area of the bars indicates that $r = 0$ and therefore that the cross-sectional area of the bars is a disc, see Figure 3.2.

d	9.5	m	load force	-890748	N
Initial A	0.2	m ²	α (for I)	0	
ρ	2768	kg m ⁻³	k	1	
E	$6.9 \cdot 10^9$	Pa	max move node	5	m
σ_{yield}	$1.72 \cdot 10^8$	N m ⁻²	min A	0.01	m ²

Table 5.1: Parameters and initial design variables for the 10-bar truss problem

The results of this test case can be found in Figures 5.2, 5.3, 5.4, 5.5 and Table 5.2. Figures 5.5 shows the absolute values of the stresses in the bars, together with their maximum value as the dotted line in the corresponding color. From this, it can be seen that the MMA and IPM method have converged and found a similar feasible solution, but the SLSQP method stops after three iterations with an infeasible solution. The SLSQP method terminates because it cannot find a negative search direction in that point anymore. Scaling the problem as with the MMA method helps a bit, but the result is still far from optimal. The feasible results look similar to the figure of shape optimization in Figure 1.3.

Another notable result is the difference in time and number of iterations for the IPM and MMA method in Table 5.2. It can be seen in Figure 5.4b that for the IPM method, the cross-sectional variables do not change much anymore after 150 iterations. This is due to the small trust-region at this point, which is why it cannot change the variables much anymore. Choosing a less accurate tolerance parameter for

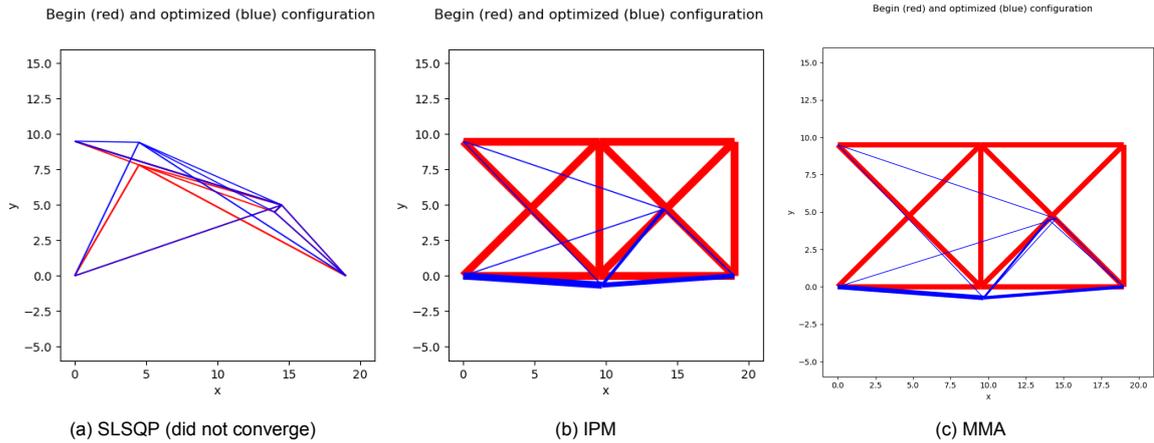


Figure 5.2: Size and shape optimization for the 10-bar truss. Initial trusses in red, optimized trusses in blue.

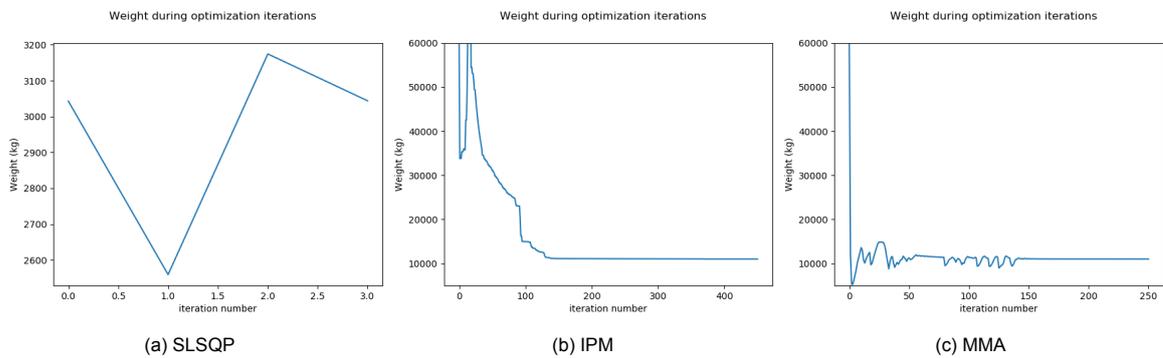


Figure 5.3: Weight of the structure during the iterations.

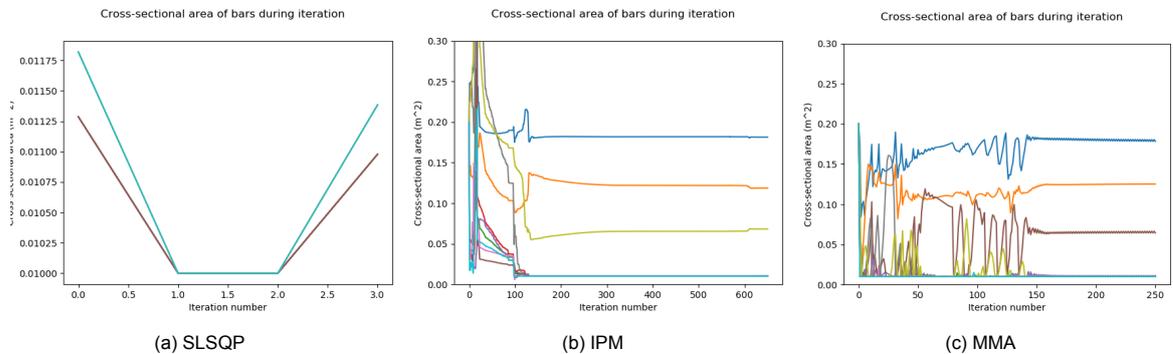


Figure 5.4: Cross-sectional areas of the bars of the structure during the iterations.

the trust region reduces this amount of steps, but also decreases the accuracy a little.

For the MMA method, the first few steps attain values for which the solution is infeasible, which can be seen in 5.3c and 5.5c. This is due to the perturbation parameter ε as explained in Section 4.5, which allows the method to search for solutions outside of the feasible region in the first steps. When this parameter becomes smaller, this is less likely to happen.

5.4. Michell's arch

For Michell's arch, the size and parameters of the structure are taken from (Wang, 2002) [30] and can be found in Table 5.3.

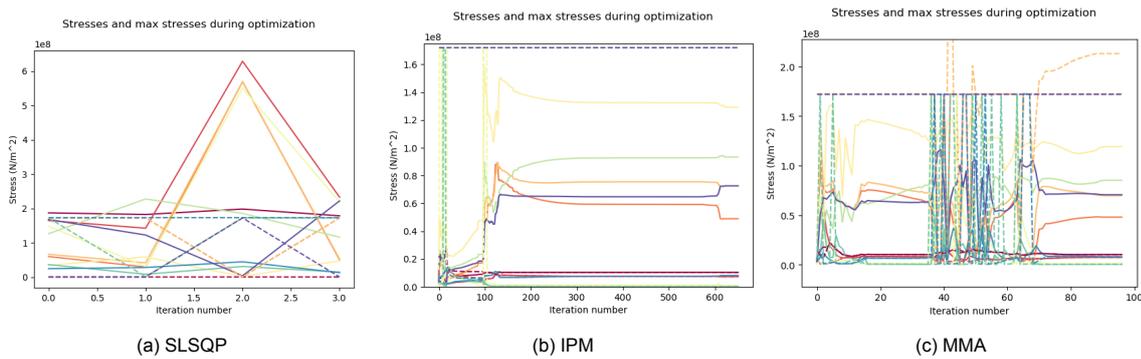


Figure 5.5: Stresses of the bars of the structure during the iterations. The dotted lines show the maximum stress for the bars in the corresponding color.

	Time (s)	Feasible	iterations	Final weight (kg)
MMA	203	yes	96	10871.37
IPM	410	yes	501	10960.04
SLSQP	1	no	3	3043.80

Table 5.2: Results for the 10-bar truss structure for the three different methods. Feasible indicates whether or not the solution is feasible.

d	0.5	m	load force	$-2 \cdot 10^6$	N
Initial A	0.05	m ²	α (for I)	0	
ρ	$7.8 \cdot 10^3$	kg m ⁻³	k	1	
E	$2.1 \cdot 10^9$	Pa	max move node	0.5	m
σ_{yield}	$2.4 \cdot 10^8$	N m ⁻²	min A	0.01	m ²

Table 5.3: Parameters and initial design variables for the Michell's arc problem

The results for this method can be found in Figures 5.6, 5.8 and 5.7. Since again, the SLSQP method produces infeasible solutions because of a positive directional derivative, only the other two methods are compared here. The final values are similar, see Table 5.4. The arch-like shape is well-known in bridge design and distributes the weight evenly over the elements. The optimal shape is therefore very intuitive here. There is a big difference in how these final values of the two methods are attained. In the first steps, the cross-sections differ much more for the IPM method, whereafter the optimal values are slowly attained. In comparison, the MMA method here does not attain values for the design variables that are outside of the limits, and the trend is more alternating than decreasing between steps 5 and 20. After that, only small changes in the design variables are made.

For the IPM method, after 60 steps the design variables are very close to the final variables. These are also compared to the final variables in Table 5.4. The last 0.48 kg reduction in weight for the IPM method takes 217 steps, which takes very long in comparison with the reduction in the first 60 iterations. If this quicker but less accurate termination is desired, the tolerance level for the IPM method can be increased. Table 5.5 shows how the minimum radius of the trust region (xtol) influences the termination of the method. It can be seen that a lower tolerance decreases the runtime drastically, without losing much accuracy for the optimum. The reduction of the tolerance level to 10^{-2} would be a good option for speeding up the method without losing too much accuracy.

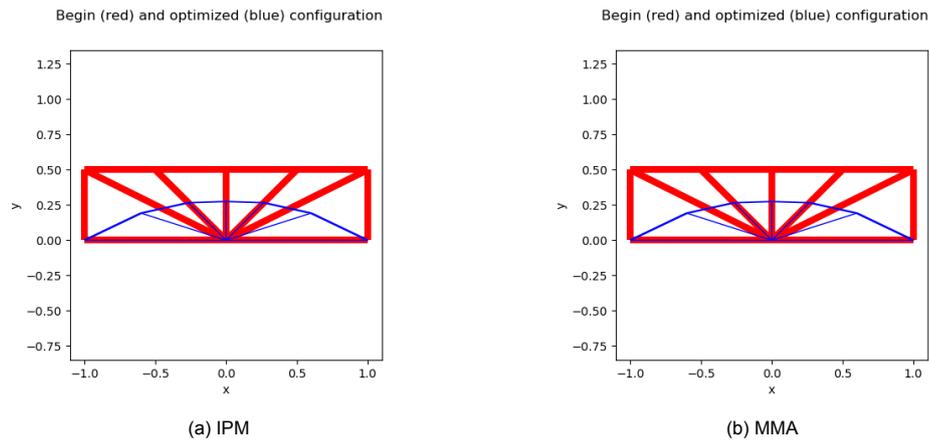


Figure 5.6: Size and shape optimization for Michell's arch. Initial trusses in red, optimized trusses in blue.

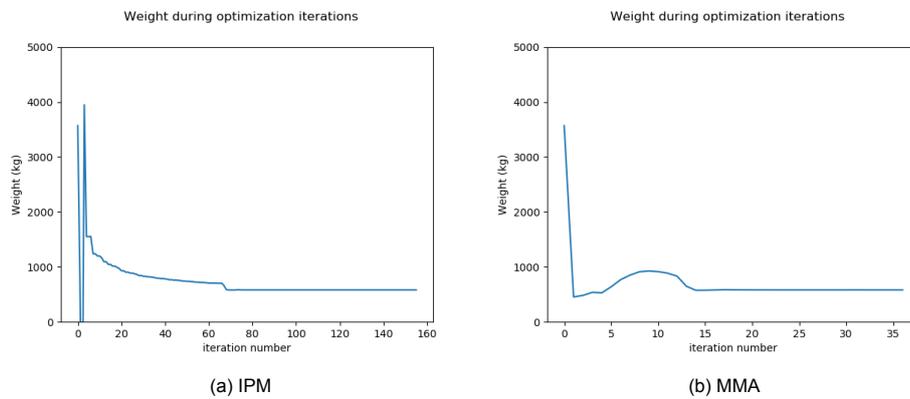


Figure 5.7: Weight of the structure during the iterations.

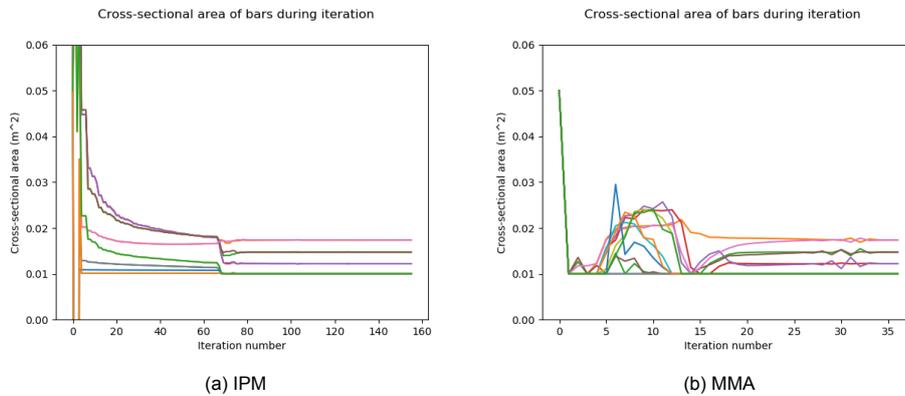


Figure 5.8: Cross-sectional areas of the bars of the structure during the iterations.

5.5. 2D Luffing boom crane

The two-dimensional luffing boom crane is a very simple crane design, which is perfect for the first test case for a crane. It is also used as a test case in ACE, which is where most of the parameters are taken from. This is done for the starting position of the nodes, the element's density, yield strength and Young's modulus, and also the load force. Here, the density, yield strength and Young's modulus are not the same for each element. The other parameters and initial values are found in table 5.6.

Next to the constraints on the stresses, the overturning moment as described in Equation (3.7) is taken as a constraint. The maximum overturning moment is taken as $6.55 \cdot 10^8 \text{ N m}$. This overturning moment constraint together with the four stress constraints on each of the bars results in five constraints

	MMA	IPM	IPM after 60 steps
Time (s)	136,96	530	109,04
Feasible	yes	yes	yes
iterations	31	277	60
Final weight (kg)	581.08	581.08	581.56

Table 5.4: Results for Michell’s arch for MMA and IPM method.

xtol value	final weight (kg)	iterations	comp. time (s)
10^{-8}	581.08	533	851.23
10^{-6}	581.09	329	569.61
10^{-4}	581.08	277	532.62
10^{-2}	581.12	176	380.85

Table 5.5: Results for different xtol parameter values for Michell’s arch, solved with the IPM method. Default value for xtol is 10^{-8} .

Initial A	0.5	m ²	max move node	10	m
α (for I)	0.9		min A	0.01	m ²
k	1				

Table 5.6: Parameters and initial design variables for the two-dimensional luffing boom crane.

in total. The cross-sectional area of each bar is a design variable, as well as the x - and y -coordinate of the hinge. This results in six design variables.

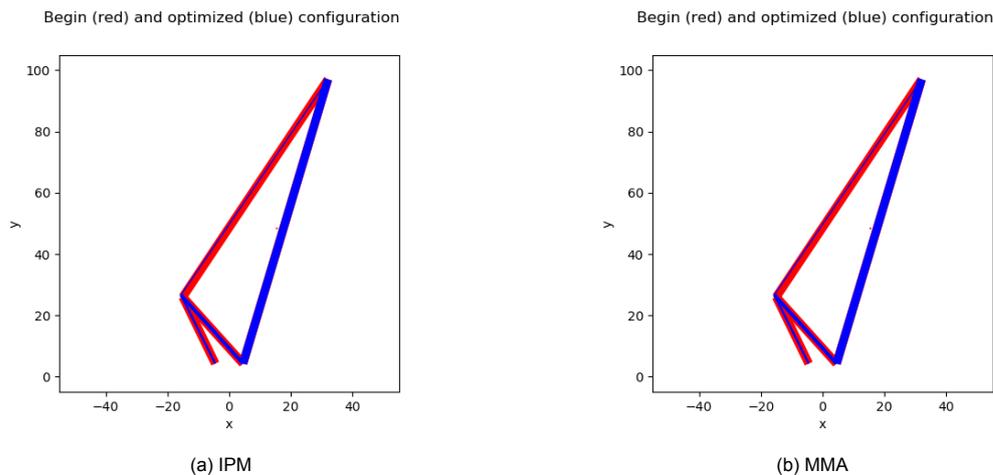


Figure 5.9: Size and shape optimization for the two-dimensional luffing boom crane. Initial trusses in red, optimized trusses in blue.

This problem is run with the methods IPM and MMA. Since Table 5.5 showed that a tolerance level of $1 \cdot 10^{-2}$ gives accurate enough results for the IPM method, this is used in the 2D luffing boom crane case. The results can be found in Figures 5.9, 5.10, 5.11, 5.12. Again, both methods give similar results and the MMA method is faster. The extra constraint on the overturning moment does not influence the optimization: Table 5.7 shows that the results are similar in this case. This means that the constraint is not active in the optimal point. It does however influence the time and number of iterations that the algorithms need.

It can also be seen in Figure 5.9 that mostly the cross-sectional area of the structure is decreased. This has the largest influence on the weight, especially in the first iterations. This means that in the first steps of the optimization, the gradient in the initial point has much higher values for the derivatives with respect to the cross-sectional areas.

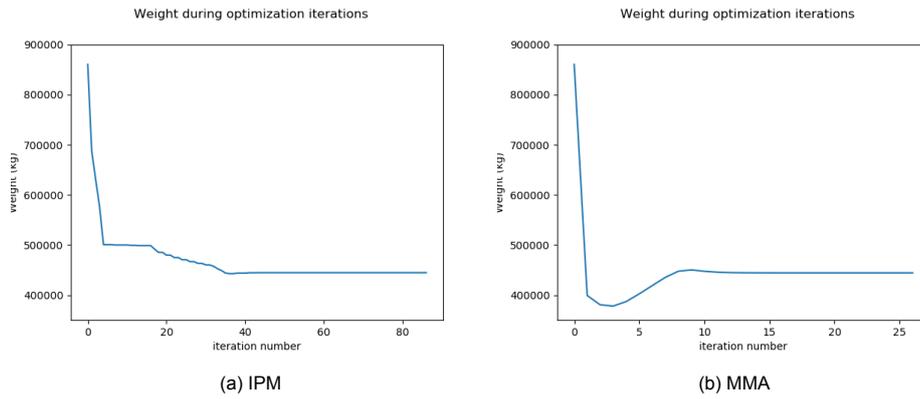


Figure 5.10: Weight of the two-dimensional luffing boom crane during the iterations.

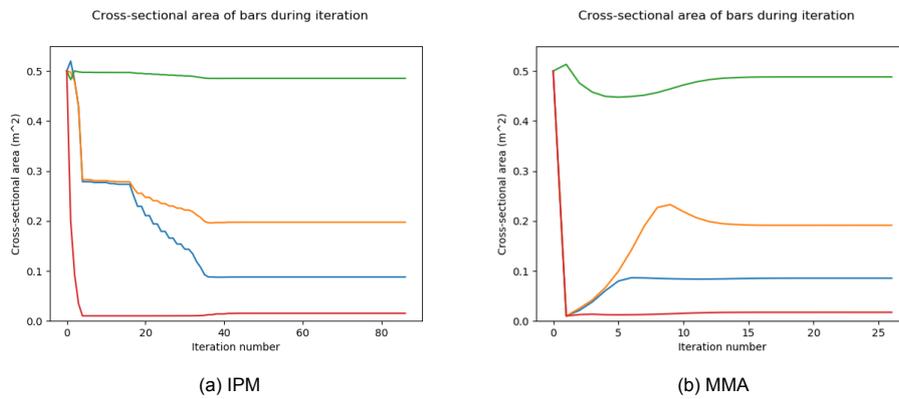


Figure 5.11: Cross-sectional area of the bars of the two-dimensional luffing boom crane during the iterations. Element 1 corresponds with blue, element 2 with yellow, element 3 with green and element 4 with red.

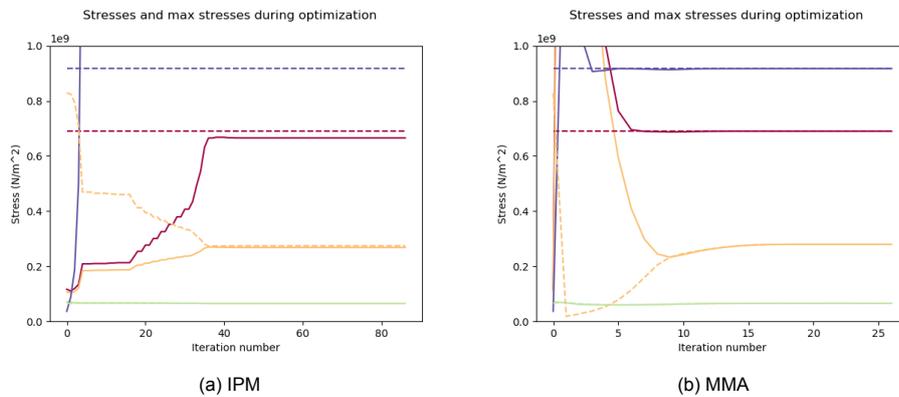


Figure 5.12: Stresses and maximal stresses during iterations for the two-dimensional luffing boom crane. Element 1 corresponds with red, element 2 with green, element 3 with yellow and element 4 with blue.

From graph 5.12b it can be seen that in the first steps, the stresses get very large and exceed the feasible values for the MMA method. This is due to too small cross-sectional areas, as can be seen in Figure 5.11b. Later, the stresses tend to obtain the maximum values. This is due to the penalty parameter that is used in this method, which is still small in the first steps and therefore allows to search for solution outside the feasible region. This is different from the trust-region method since in 5.12a it can be seen that the maximum stresses are not exceeded that much.

	<i>with overturning moment constraint</i>		<i>without overturning moment constraint</i>	
	MMA	IPM	MMA	IPM
Time (s)	17.7	25.4	34.15	46.4575
Feasible	yes	yes	yes	yes
Iterations	26	87	19	264
Weight (kg)	444382.05	444382.05	444382.06	444382.05
node1	(-15.612, 26.645)	(-15.613, 26.645)	(-15.613, 26.645)	(-15.613, 26.645)
A1 (m²)	0.07760	0.07759	0.07759	0.07759
A2 (m²)	0.1754	0.1754	0.1754	0.1754
A3 (m²)	0.46576	0.46576	0.46576	0.46576
A4 (m²)	0.01611	0.01612	0.01612	0.01612

Table 5.7: Table with final values for the optimization of the two-dimensional luffing boom crane constrained on stresses and overturning moment.

5.6. 2D Tetrahedron crane

For the optimization of the two-dimensional Tetrahedron crane, more constraints than just on stress have to be added to retrieve a useful solution. This is because the structure has three nodes that are free variables, which are restricted to some of the constraints that are elaborated upon in Chapter 2. The initial shape of the crane can be seen in Figure 5.1d, and the parameters and initial cross-sections can be found in Table 5.8.

Initial A	0.5	m ²	max move node	40	m
α (for I)	0.95		min A	0.01	m ²
k	1				

Table 5.8: Parameters and initial design variables for the two-dimensional Tetrahedron crane.

The nodes that are free are the hinge, the heel and the mast top, see also Figure 2.3. Next to the constraints on stresses, shape of the crane is constrained by

- The maximum overturning moment M_O^{max} as described in Equation (3.7). The maximum overturning moment in the design point is set to $6.55 \cdot 10^8$ N m.
- The minimal distance between the mast and the heel, as described in Equation (3.11). This is 9 meters.
- The heel position, which does not touch the mast in both the lowest and highest position. Therefore, the heel has to be on the right side of the mast in both positions, see Equation (3.14).
- The heel position, which should always be below the mast top.
- The minimum boom clearance as described in Equation (3.12). At a height of 107 meters, the minimal distance from the load line of the load to the jib is at least 12 meters.

In total, this problem has 14 design variables and 15 constraints, whereas for the luffing boom crane, there were less constraints than design variables. The mathematical formulation for these constraints can be found in Chapter 3. Most of these extra constraints do not have to be scaled since their values are already between 1 and 100, except for the overturning moment. This is scaled as

$$\frac{M_O - M_O^{max}}{10^7} \leq 0.$$

The results for the two-dimensional Tetrahedron crane for the MMA and the IPM can be found in Figures 5.13 - 5.16 and Table 5.9. Here, a difference in the results of the two methods can be seen. The IPM method does not converge if the starting values as indicated in Table 5.8 are used. This is because these initial design variables are further away from the optimum than before, and the trust region is decreasing too quickly for the method to reach the optimum. The tolerance level is reached before the constraints are satisfied, which results in an infeasible solution. This can be solved by

	MMA	IPM
Time (s)	49.73	457.5
Feasible	yes	yes
Iterations	28	596
Weight (kg)	$5.3533 \cdot 10^5$	$5.12107 \cdot 10^5$
hinge x-coordinate (m)	-15.414	-5.984
hinge y-coordinate (m)	43.69	62.64
mast top x-coordinate (m)	4.11	4.23
mast top y-coordinate (m)	70.48	71.85
heel x-coordinate (m)	15.48	15.86
heel z-coordinate (m)	65.28	66.54
back leg A (m ²)	0.15	0.14
front leg A (m ²)	0.15	0.01
mast A (m ²)	0.27	0.35
mast spreader A (m ²)	0.057	0.096
jib spreader A (m ²)	0.14	0.099
pendant A (m ²)	0.025	0.024
hoist A (m ²)	0.029	0.060
jib A (m ²)	0.35	0.34

Table 5.9: Results of the size and shape optimization of the two-dimensional Tetrahedron crane.

choosing better initial design variables. Since the optimal values are known from the MMA method, this can be done, but it is of course inefficient if it is necessary to run the optimization twice with different methods. The results for the IPM method found by replacing the initial cross-sectional areas of 0.5 m² with

A1	A2	A3	A4	A5	A6	A7	A8
0.2	0.2	0.3	0.05	0.1	0.03	0.03	0.4.

Here, the numbering of the elements corresponds with Figure 5.1c. Even when the initial design variables are very close, the method still takes a long time computing. Eventually, a solution is found, but this time it is different from the MMA optimum and the final weight is less than the final weight of the MMA optimization, see Table 5.9. Especially the stresses in the elements 2 and 5, the front supports and the jib spreaders, are very different, as can be seen in Figure 5.16. It makes sense that the stresses that are different are the compressive stresses, since for the tensile stresses, the maximum $\sigma_{\text{yield}}^{\text{max}}$ is a constant which is attained for both the MMA and IPM method.

Even though the result from the IPM method is feasible according to the constraints, it is not a practical solution. This is because there is no boom clearance at the minimum hoist radius, so nothing can be lifted at this position, see Figure 5.17a. However, this is not implemented as a constraint.

As can be seen in Figure 5.13, the base of the frame gets higher than the original configuration, and in Figure 5.17 it can be seen the maximum limit of the y -coordinate of the heel is reached, since it is at the same height as the mast top. Due to these findings, Tetrahedron thought of the constraint on the height of the base frame that is described in Section 3.4 in Equation (3.8).

In addition, the hinge is closer to the crane than in the original configuration. This is because the only constraint that influences this is the overturning moment, and the increase of overturning moment is compensated with a shorter jib, which is the element with the largest cross-sectional area and is also very long. This element therefore has a large mass and thus a big influence on the center of gravity of the structure. When the side-lead and off-lead are included to the load vector, the hinges would also likely move away from the origin.

As can be seen by the examples in this chapter, both the IPM and the MMA method attain an optimum for all two-dimensional experiments. Most often it is the same configuration, except for

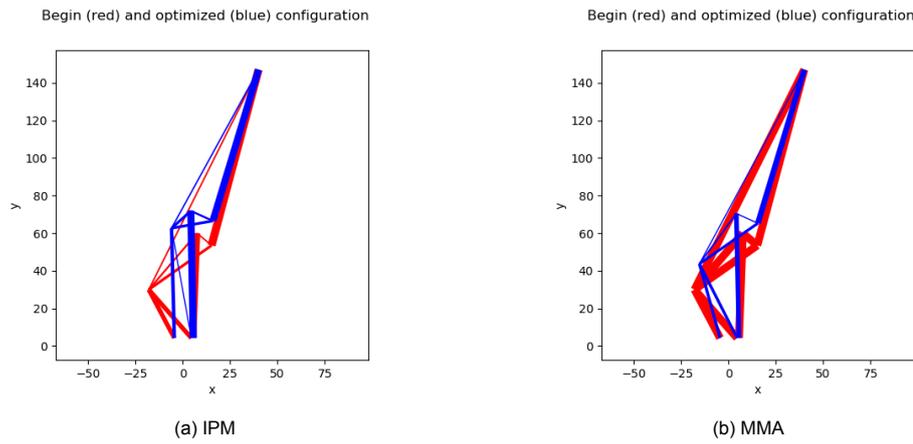


Figure 5.13: Size and shape optimization for the two-dimensional Tetrahedron crane. Initial trusses in red, optimized trusses in blue.

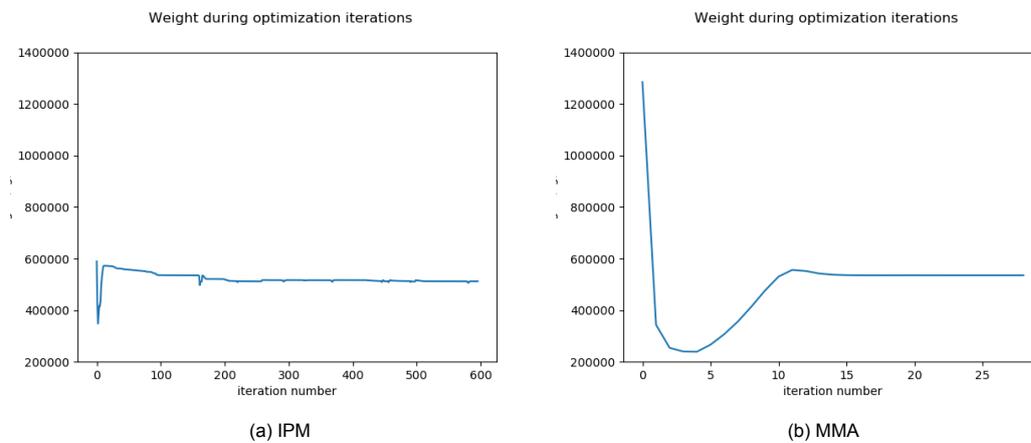


Figure 5.14: Weight of the two-dimensional Tetrahedron crane during the iterations.

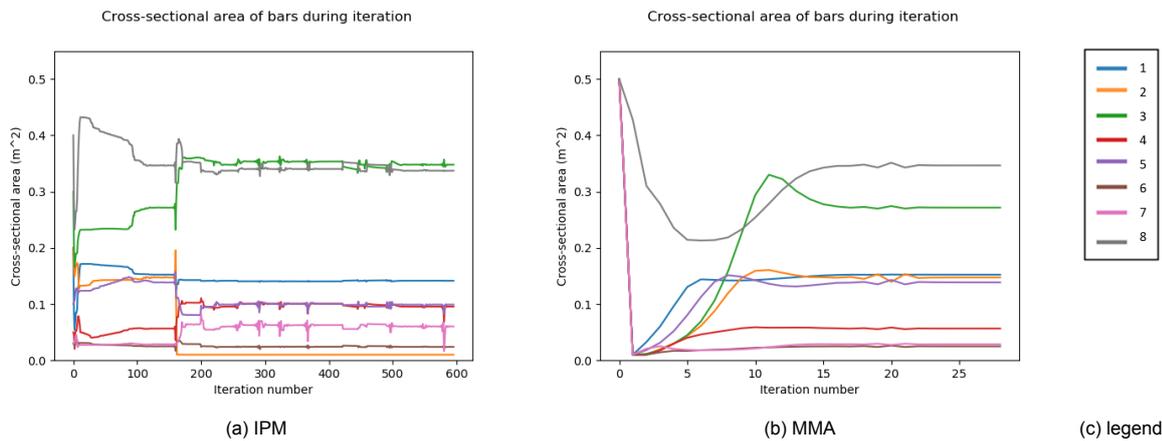


Figure 5.15: Cross-sectional areas of the bars during the iterations. The numbers of the legend correspond with the numbers in Figure 5.1d.

the Tetrahedron crane test case. For the 10-bar truss problem and Michell’s arch, the results also correspond with the literature. IPM takes a much longer time to reach the final weight with the default tolerance level 10^{-8} , but the computation time can be drastically reduced if the tolerance level is increased to 10^{-2} . The decrease in accuracy is then still small, and insignificant on the scale of a crane structure. Therefore, both methods will be tested on the three-dimensional Tetrahedron crane

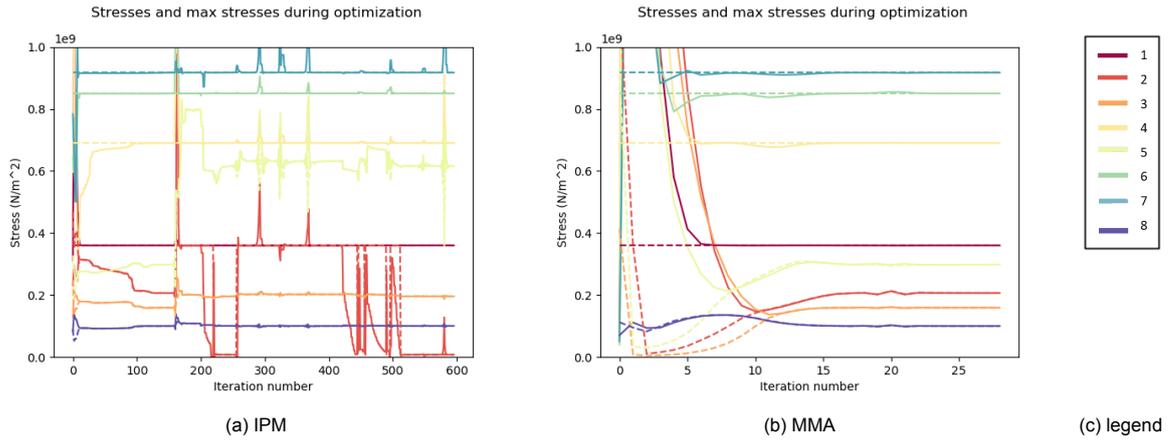


Figure 5.16: Stresses and maximal stresses during iterations for the two-dimensional Tetrahedron crane. The numbers of the legend correspond with the numbers in Figure 5.1d.

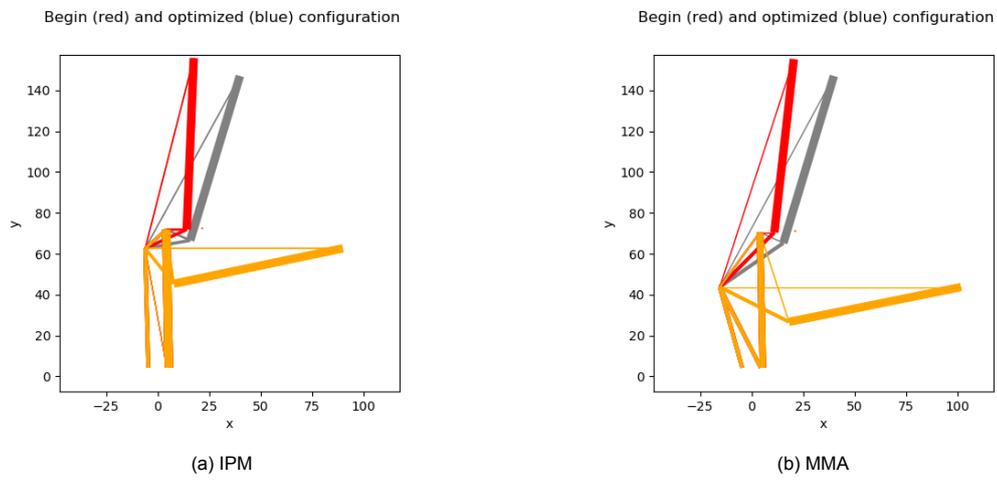


Figure 5.17: Crane in design point, highest and lowest hoist radius position. The crane is set to be horizontal in the lowest position, and to not touch the mast in the highest position.

problem.

6

3D experiment: Tetrahedron crane

In this chapter, the results of the three-dimensional Tetrahedron crane are presented and discussed. This does not only include a comparison between MMA and IPM, but also an insight in how the alteration of certain constraint parameters influence the results. Finally, some convergence results are discussed as well.

6.1. Problem definition

For this final problem, all constraints that are elaborated upon in Chapter 3 are used for the optimization of the 3D Tetrahedron crane. The parameters are listed in Tables 6.1, 6.2 and 6.3. The initial values come from Tetrahedron, and the parameter values are also selected in consultation with Tetrahedron.

node	x-coordinate (m)	y-coordinate (m)	z-coordinate (m)	additional weight (kg)
Supports back	-4.5	4.5	4.25	225761
Supports front	4.5	4.5	4.25	$1.425 \cdot 10^5$
Support center	5.5	0	4.25	$1 \cdot 10^4$
Hinge	-17	15	34.25	60683
Mast top	7.5	0	64.25	$6.5 \cdot 10^4$
Heel	15.18	0	53.51	$6.5 \cdot 10^4$
Top	40	0	151.25	$2.6 \cdot 10^5$

Table 6.1: Initial positions and additional weight at each node. For symmetric nodes, the coordinates indicate the right node.

Edge	Initial A (m ²)	ρ (kg ³ m ⁻¹)	E (Pa)	σ_{yield} (N m ⁻²)
Jib	0.168	7850	$210 \cdot 10^9 \cdot 0.77$	$690 \cdot 10^6$
Pendant	0.05	1400	$116 \cdot 10^9$	$850 \cdot 10^6$
Hoist	0.0902	6366	$100 \cdot 10^9 \cdot 0.7$	$917 \cdot 10^6$
Jib spreader	0.0766	7850	$210 \cdot 10^9 \cdot 0.95$	$360 \cdot 10^6$
Front leg	0.177	7850	$210 \cdot 10^9 \cdot 0.85$	$360 \cdot 10^6$
Back leg	0.207	7850	$210 \cdot 10^9$	$360 \cdot 10^6$
Mast	0.16	7850	$210 \cdot 10^9 \cdot 0.8$	$690 \cdot 10^6$
Mast spreader	0.0333	7850	$210 \cdot 10^9$	$690 \cdot 10^6$
Back cross	0.0517	7850	$210 \cdot 10^9 \cdot 0.85$	$360 \cdot 10^6$
Transverse	0.0459	7850	$210 \cdot 10^9 \cdot 0.5$	$360 \cdot 10^6$

Table 6.2: Initial cross-sectional areas, densities, Young's moduli and yield strength for all edges. For symmetric elements, the values are applicable to both members.

If any of the values is altered for the results, this will be indicated.

Parameter	value
r_{leg}	2.5 m
r_{platform}	7.05 m
F_{load}	$1.25 \cdot 10^3 \cdot 9.81 \cdot 1.1$ N
off-lead	1 degree
side-lead	3 degrees
M_o^{max}	$6.55 \cdot 10^8$ Nm
d^{max}	22.5 m
d_{hoist}	7 m
$d_{\text{boom clearance}}^{\text{min}}$	11 m
h_{load}	102.75 m
max hoist radius	100 m
min hoist radius	20 m
M_z^{max}	$9.684 \cdot 10^6$
α	0.98
k	1

Table 6.3: Parameter values for the optimization problem. All parameters are explained in Chapter 3.

6.2. Comparison new method with existing optimization method

The results for the optimization of the 3-dimensional Tetrahedron crane with the Method of Moving Asymptotes are discussed in this section, and compared to the already existing optimization method. A comparison is made between the methods' computation time, the amount of weight reduction and the robustness of the methods. To make a good comparison, the existing optimization method will first be explained.

The optimization method that is implemented in ACE, minimizes the weight of the structure with the cross-sectional areas of the bars as design variables. The method follows the following steps:

- Step 1: Set sufficiency factor $p \in (0, 1)$, sufficiency step Δp , max sufficiency factor P , max forces of edges λ^{max} , cross-sectional area decrease step $s \in (0, 1)$.
- Step 2: Compute the forces in the elements of the crane λ_b , under a discretization of all positions of the crane.
- Step 3: For each bar, it is checked if the force is close enough to the maximum force, as

$$|\lambda_b|_{\text{max}} \in [p \cdot |\lambda^{\text{max}}|, |\lambda^{\text{max}}|].$$

If this is not the case, there can be two situations:

- $|\lambda_b|_{\text{max}} < p \cdot |\lambda^{\text{max}}|$:
Update the cross-sectional area of b as

$$A_b := A_b - s \cdot (A_b - A_b^{\text{min}}).$$

Here, A_b^{min} is the minimal cross-sectional area that would be able to withstand the force λ^{max} .

- $|\lambda_b|_{\text{max}} > |\lambda^{\text{max}}|$:
Return to previous cross-sectional areas, and reduce the cross-sectional area decrease step with

$$s := s^2.$$

Step 4: If the forces are close enough to λ^{max} , update $p := p + \Delta p$ and repeat step 2-3.

Step 5: Repeat Step 2-4 until $p > P$.

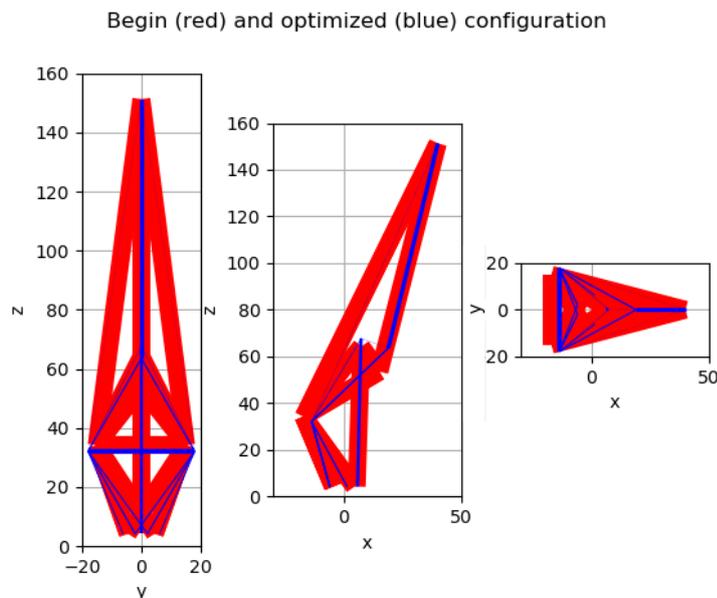


Figure 6.1: Result for the new design of the Tetrahedron crane after size and shape optimization with the method of moving asymptotes.

An advantage of this model is that it computes the forces in all positions. However, it does not always converge to a solution if the sufficiency factor or the step size for adjusting the cross-sections are not chosen well. The MMA and IPM method also needs proper scaling of the constraints to find a good solution, but the scaling is not very sensitive. Next to that, the new model also includes shape optimization, which is a big advantage over the already existing method. Also taking the Euler buckling into account is another advantage with respect to the existing method.

The graph of the resulting crane of the MMA method can be found in Figure 6.1 and the comparison with the ACE method can be found in Table 6.4. The methods are compared for starting values of $A = 1.0\text{m}^2$ for all bars, which is why the original crane in Figure 6.1 has very thick elements. From Table 6.4 it can be seen that the MMA method is faster than the existing method, and needs about half the amount of iterations in comparison with the ACE method. And, more importantly, the weight is reduced more.

For the IPM, the method takes a very long time. This could be expected since it is also slower than MMA in the test cases. More importantly, the method does not find a solution from this starting point, which is very far from the feasible set. The constraints on the stresses and overturning moment are never satisfied during the iterations, and as can be seen in Table 6.4, some of the cross-sectional areas of the bars are negative, which is also a violation of the bounds. A good suggestion to reduce the computation time of the IPM, is to first run the MMA method and find values that are close to the optimum, and then use IPM to refine the solution. Therefore, it is important to also test the new methods with more reasonable initial cross-sectional areas as defined in Table 6.2. But even then, the method does not find a feasible solution. Therefore, only the results of the MMA method will be discussed in more detail on convergence, robustness and parameter sensitivity.

In Table 6.5, it can be seen which constraints are active at the optimum of the MMA method with the configuration as stated in Section 6.1. Most of the stress constraints are active, and if the stress constraint of an element is not active, then it is active for its symmetric counterpart. This makes sense, since because there is a side-lead, the symmetric elements are not loaded equally. Another thing that is interesting, is that the constraint that the heel should not touch the mast in the bottom position is never attained in any test configuration. This suggests that this constraint is unnecessary. This does not hold for any other constraint that is inactive in the situation of Table (6.5), since they are active when the problem has different parameters.

	IPM (infeasible)	MMA	ACE
Computation time (s)	1726	231	548
Number of iterations	523	27	47
Starting weight (kg)	$4.9693 \cdot 10^6$	$4.9693 \cdot 10^6$	$4.9693 \cdot 10^6$
Final weight (kg)	$6.9502 \cdot 10^5$	$5.3109 \cdot 10^5$	$5.62 \cdot 10^5$
back leg x-coordinate (m)	-3.26409	-5.90104	-4.5
back leg y-coordinate (m)	-5.15159	2.08554	4.5
front leg x-coordinate (m)	5.70189	1.4079	4.5
front leg y-coordinate (m)	-2.39596	6.15855	4.5
hinge x-coordinate (m)	-19.0078	-13.64815	-17
hinge y-coordinate (m)	12.0686	17.88793	15
hinge z-coordinate (m)	25.27943	32.13689	34.25
center leg x-coordinate (m)	3.62239	5.87866	5.5
mast top x-coordinate (m)	10.281	7.38863	7.5
mast top z-coordinate (m)	63.36764	67.724971	64.25
heel x-coordinate (m)	9.83439	18.96655	15.1789
heel z-coordinate (m)	36.58708	63.58295	53.505
back leg A (m ²)	0.11529	0.07789	0.202
front leg A (m ²)	0.09546	0.06676	0.168
mast A (m ²)	0.1539	0.17069	0.163
transverse A (m ²)	1.28206	0.264737	0.0483
mast spreader A (m ²)	-0.01229	0.02345	0.0337
jib spreader A (m ²)	-0.00665	0.08262	0.0799
pendant A (m ²)	-0.01698	0.008	0.051
hoist A (m ²)	0.02228	0.01414	0.0926
jib A (m ²)	0.31909	0.23328	0.172
back cross A (m ²)	-0.00165	0.08562	0.0479

Table 6.4: Results of the size and shape optimization and of the existing size optimization implemented in ACE.

Constraint	Active?	Element (stress)	Active?
overturning moment	yes	back support right	yes
heel doesn't touch mast top position	yes	back support left	no
heel doesn't touch mast bottom position	no	front support right	yes
min mast heel distance	yes	front support left	no
heel below mast at top	no	mast	yes
min boom clearance	no	transverse	yes
jib spreader distance from leg	no	mast spreader right	yes
mast spreader distance from leg	yes	mast spreader left	yes
support front distance from leg	yes	jib spreader right	yes
distance hinge to other legs	yes	jib spreader left	yes
max moment times weight base frame	no	pendant right	no
support back to leg distance	no	pendant left	yes (minimum)
support front to leg distance	no	hoist	yes
support center to leg distance	no	jib	yes
support back to outside slew platform	yes	back cross right	no
support front to outside slew platform	no	back cross left	yes
support center to outside slew platform	yes		

Table 6.5: Table showing all constraints and whether or not, at the found optimum with the MMA method, the constraint is active. The right side shows the elements and whether or not the stress constraint is active, and on the left, the results for the other constraints are given. For the pendants, it is also checked if the minimum pendant stress of Equation (3.5) is attained.

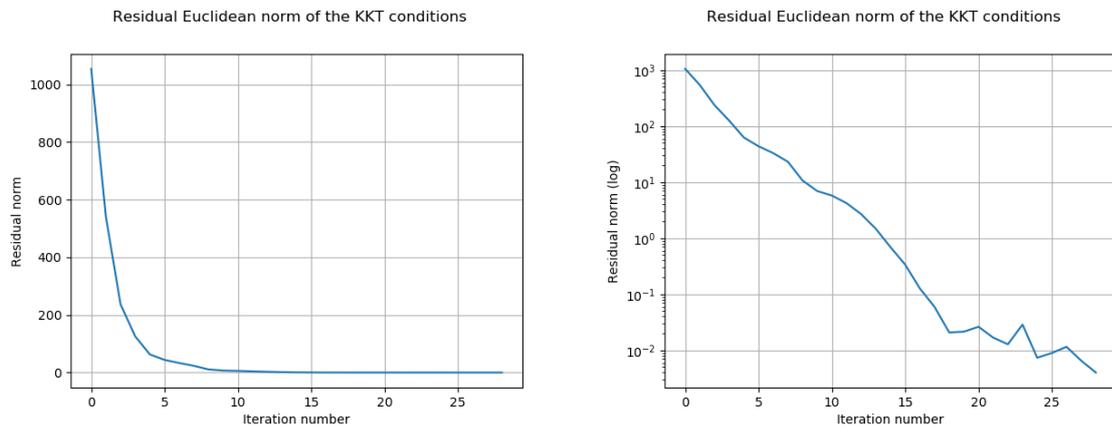


Figure 6.2: Graph of the norm of the residuals of the KKT conditions per iterations for the Method of Moving Asymptotes. Left is in linear scale, right is in semi-log scale.

6.2.1. Convergence

Figure 6.2 show the evolution of the Euclidean norm of the residuals $\|\delta(w)\|_2$ of the perturbed KKT conditions for the MMA method. This norm $\|\delta(w)\|_2$ converges to 0 as $\epsilon \rightarrow 0$, if an optimum is found. The method is set to terminate if $\|\delta(w)\|_2 < 0.005$. From these figures, it can be seen that the method converges linearly. The rate of convergence is approximately 0.7. The linear convergence is expected, since the method applies a first-order quasi-linear approximation of the constraints and objective function.

It is interesting to see that in some steps, the residual norm increases, while the overall trend is a decrease. This only happens when the residual is small, lower than 0.05. This might happen when ϵ decreases, and the decrease of the other terms in Equations (4.51)-(4.54) is less.

6.2.2. Robustness

When using the MMA method with initial cross-sectional areas of 1 m², the first few iterations result in an ill-conditioned matrix in the primal-dual system, with a condition number of around 10¹⁸. This is

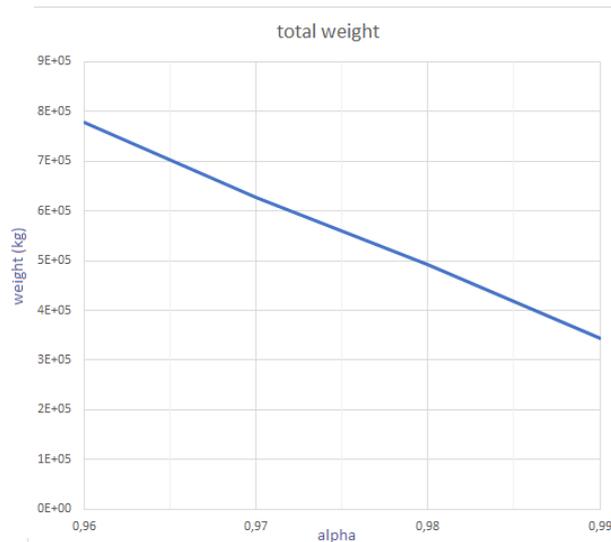


Figure 6.3: Optimized weight of Tetrahedron structure for different values of α .

because the scaling of the method only works for values close enough to the solution. However, within 3 iterations, the solution becomes close enough and the condition number of the matrix decreases to an acceptable number of 10^6 , and decreases even further afterwards. If the MMA is used with the initial values close enough, such as in Table 6.2, there is no such problem as long as the scaling of the objective function and constraints is done properly, as explained in Section 5.1.

As mentioned in the beginning of this section, the existing method in ACE sometimes does not converge if the sufficiency factor p or the step size for adjusting the cross-sectional areas s are not chosen well. The best way of finding a good sufficiency factor is with trial and error, whereas for the MMA method, it is known that the effects of ill-conditioning are reduced with proper scaling.

6.3. Parameter sensitivity

It is interesting to see how the solution changes for different values of parameters. This shows how sensitive the problem is to changes, and also gives an insight in the mechanical properties of the crane. The parameters that are altered in this Section are the bar thickness and the side-lead. This is because these parameters are expected to change the shape of the crane a lot, and the change can be explained through the mechanical aspects of the crane.

α	0.95	0.96	0.97	0.98	0.99
I (A=0.1)	0.0155	0.0195	0.0261	0.0394	0.0792
I (A=0.05)	0.00388	0.00488	0.00653	0.00985	0.0198

Table 6.6: Values of moment of inertia for different bar thickness α . The cross-sectional area of the bars is taken as 0.1 and 0.05.

6.3.1. Bar thickness

This parameter is chosen as an assumption on the thickness of the bars. A reasonable value for the Tetrahedron crane would be $\alpha \in (0.96, 0.99)$, since the bars of the structure would be very thin. The other parameters remain as described in Tables 6.1-6.3. The influence of α on the area moment of inertia of bars can be seen in Table 6.6. A larger area moment of inertia will result in a higher critical force for the bars under pressure. This means the bars under pressure can have a smaller cross-sectional area without failing if α is chosen larger.

The element with the most compression is the jib, so for this case it is interesting to see how its cross-sectional area differs for values of α . If the cross-sectional area of the jib is larger, then the other

α	0,96	0,97	0,98	0,99
total weight (kg)	$7.78 \cdot 10^5$	$6.27 \cdot 10^5$	$4.92 \cdot 10^5$	$3.45 \cdot 10^5$
A jib (m ²)	$3.78 \cdot 10^{-1}$	$2.99 \cdot 10^{-1}$	$2.45 \cdot 10^{-1}$	$1.82 \cdot 10^{-1}$
A transverse (m ²)	$4.51 \cdot 10^{-1}$	$1.42 \cdot 10^{-1}$	$3.64 \cdot 10^{-2}$	$2.34 \cdot 10^{-2}$

Table 6.7: Final weight and cross-sectional areas of the jib and transverse of the Tetrahedron crane for different values of α .

elements have to compensate more for this weight to reduce the overturning moment. Therefore, the weight of the structure is higher when α is smaller. This can be seen in Table 6.7 and Figure 6.3. There is actually a linear relationship between the final weight and α .

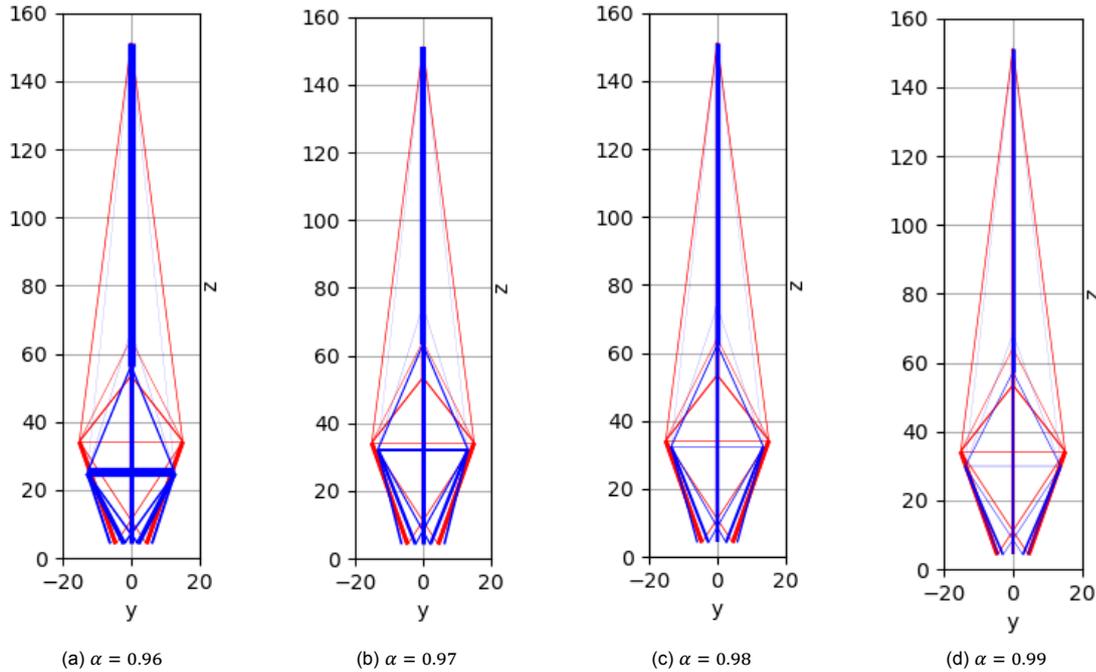


Figure 6.4: Graphs showing the yz-plane of the crane in final configuration for different α .

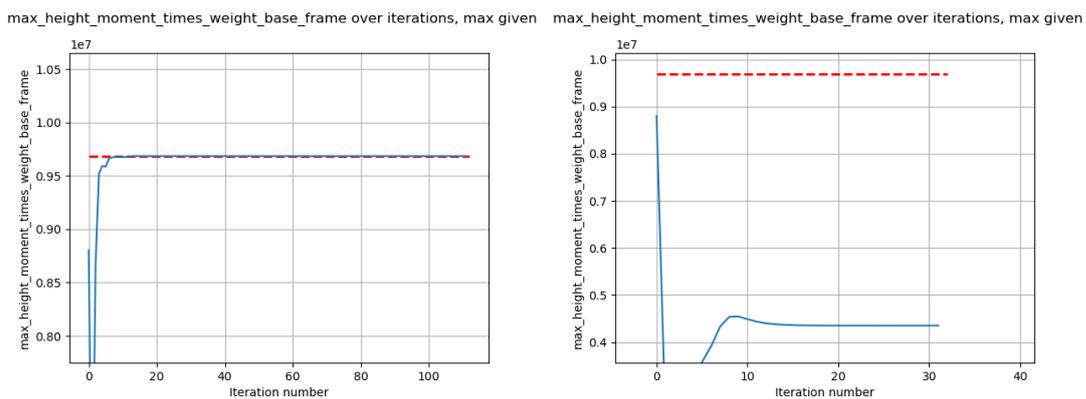


Figure 6.5: Graphs showing the constraint on base frame height times weight as in (3.8). The red line shows M_z^{max} and the blue line is the left-hand side of Equation (3.8). The left graph shows the result for $\alpha = 0.96$, the right graph shows the result for $\alpha = 0.99$.

It is also interesting to see the difference in the values for Constraint (3.8). The values for this constraint for $\alpha = 0.96$ and $\alpha = 0.99$ can be found in Figure 6.5. For a smaller α , the maximum height

times weight of the base frame is attained. The cross-sectional area of the transverse has to be much larger in this case, which is why the weight of the base frame is much larger. To compensate this, the hinges are positioned very low, see Figure 6.4. For $\alpha = 0.99$, the cross-sectional areas of the bars can become so small that the weight of the base frame is never large enough to reach the maximum moment M_z^{max} .

6.3.2. Side-lead

Another interesting factor that influences the crane shape is the side-lead. If the side-lead is higher, the y -component of the external force is larger. To compensate this force, it is expected that the crane becomes wider, i.e. the y -coordinate of the hinges gets larger. If the side-lead is 0, the crane should become as small as possible without touching the jackup leg. The results of this test can be found in Figure 6.6 and Table 6.8. It can be seen that indeed, the hinges are right above the supports if there is no side-lead, and they are further away than in the initial configuration if the side-lead is 9 degrees, which is much higher than the angle that Tetrahedron uses for their calculations.

side lead (degrees)	0	3	6	9
Total weight (kg)	$3.91 \cdot 10^5$	$4.92 \cdot 10^5$	$5.64 \cdot 10^5$	$6.42 \cdot 10^5$
y-coordinate hinge (m)	9.9	13.4	15.6	17.0

Table 6.8: Resulting weight and y-coordinate of the hinge for different side-lead angles.

The y -coordinate of the hinge increases for larger side-lead, as expected. Constraint (3.10) on the distance of the hinge to the other legs is active for all different values of the side-lead. This means that for smaller y -coordinates of the hinge, the x -coordinate is larger such that the distance of the hinge to the origin in the xy -plane is still the same.

The weight of the structure increases for larger side-lead angles. This is because the total force on the structure increases when the angle is larger, and more importantly, there are more forces in the horizontal direction. Since the elements move outwards to handle these forces, the element length increases, which increases the total weight. The increase of weight is approximately 10^5 kg for each 3 degrees of side-lead that is added.

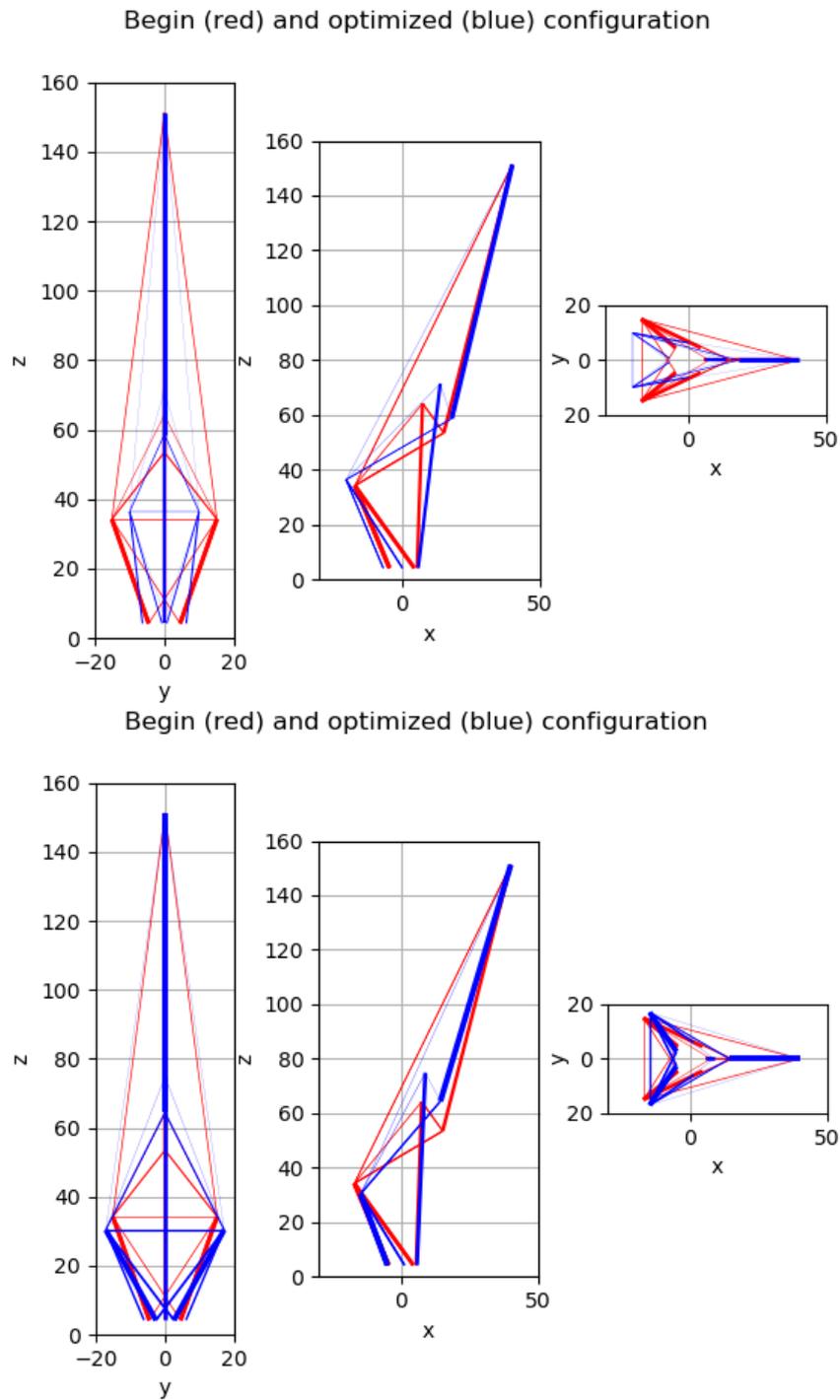


Figure 6.6: Graphs showing the shape and sizes of the elements for different side-lead angles. On the top, the side-lead angle is 0 degrees, and below this angle is 9 degrees.



Discussion and recommendations

In this chapter, the results are summarized and discussed, and some recommendations for further research in this field are made.

The goal of this research was to reduce the weight of the Tetrahedron structure with a combined shape and size optimization method. To reach this goal, firstly the formulation of the crane optimization problem is studied thoroughly and a minimization problem is formulated that described the design space of the crane in a mathematical manner. All physical and mechanical properties that have to be taken into account are described as constraints on the design variables.

Next to that, a study on different optimization methods is performed to research which method would be best suited for the crane optimization problem. The methods are also tested in different cases, and the Method of Moving asymptotes is found to be the most promising method. The resulting code is written such that it can easily be implemented in the existing program ACE.

The optimization method is now only tested on one configuration for the 3-dimensional Tetrahedron crane, for which Tetrahedron has performed many design iterations to improve the structure before the crane parameters were final. Therefore the design of the Tetrahedron crane as described in Chapter 6 is already close to the optimum.

The newly developed optimization method will be of even more importance when Tetrahedron starts the design of a new Tetrahedron crane for a different type of ship or load, which changes the input parameters drastically. Then, instead of having to start designing from scratch again, the optimization model will give them a very good starting point from which the new crane design can be made. This saves Tetrahedron much time and is therefore very valuable for the company.

From the results of Chapter 5 and 6, the Method of Moving Asymptotes seems the most promising for the optimization of crane structures. The SLSQP method is very sensitive to starting points and scaling, which almost always gave infeasible solutions and therefore useless results. This method was therefore dropped after the first 2D experiments and only the other two methods, MMA and IPM, were tested on the 3-dimensional Tetrahedron crane optimization problem.

The interior-point method converged for most of the test cases without applying any scaling, and could therefore be said to be a robust method. However, when the problem size increases for the 3-dimensional Tetrahedron crane, the method fails more often than MMA, even with the same scaling applied. This might be because the difference between the number of constraints and the number of design variables is much larger for this case. This would be in line with the literature, since here it is suggested that the IPM method performs well for problems with a large amount of free variables [16], which is not the case for the crane optimization problem.

The IPM method takes a considerably longer time for the test cases as well. Since the 3-dimensional problem is even larger, even if the method would converge, MMA should be preferred because of the computation time. It was already expected that the MMA is faster than the IPM method in the comparison in Chapter 4, and from the results of Chapter 5. MMA only applies first-order approximations, whereas for the IPM method, the computation of the approximation of the Hessians takes a considerable

long time. For the MMA method, if the scaling of the objective function and constraints is done as explained in Section 5.1, the method converges linearly to an optimum in reasonable time. In Table 4.1, it was however suggested that the Interior point method would have a better convergence success rate than MMA. This might still hold for general nonlinear optimization methods, but for structural optimization problems discussed in Chapters 5 and 6, MMA outperforms IPM. The nonlinearity of the objective function likely plays a role here, since the second-order approximation of the objective function for the IPM method is more precise than the approximation of the MMA method, as was also suggested by Schittkowski and Zillober in [28]. The objective function of structural weight is not very nonlinear, which is why the approximation of MMA is sufficient.

Another advantage of MMA is that its parameters are easier to alter since the code is freely available, whereas the IPM method is part of the *scipy* package and is therefore a black box method.

The recommendations can be split into two categories. Firstly, the recommendations on the formulation of the problem will be presented after which recommendations on the optimization methods and the Python code will be given.

Recommendations - problem formulation

The optimization formulation is of course a simplified model, and can be extended with other mechanical principles if found necessary. One example of this is that for now, the cross-sectional area of the bar is considered an annulus with a certain thickness. It might be useful to also consider the parameter α which determines this thickness in Equation (3.3), as a design parameter. The formulation might even be extended with a selection of different shapes for the cross-sectional area, such as a box or ellipse. This would change the optimization problem drastically, however, since this would be a discrete instead of a continuous design variable. The chosen optimization methods would be no longer applicable to the problem then.

Another mechanical principle that is neglected in the model right now, is local buckling of the bars. This is an effect that takes place in thin plates, that locally deforms the plate which results in a failure. Local buckling might especially be interesting to check if the annulus thickness and/or shape should be taken as design variables.

One of the differences between the existing model and the new optimization model, is that in the new model the stresses are only computed in the design point, whereas in the existing model this is done for all hoist radii. It could be checked what the forces are like in the other positions, but for positions lower than the design point, the external load has to be reduced as well. Instead of making these computations, it is now chosen to compute the load curve of the new crane, which indicates what is the max external load that the crane can carry in each position. This is useful for Tetrahedron, since the load curve is a good measure of a crane's performance. If the model should be extended to also compute the stresses and maybe also other constraints for different hoist radii, this should be done by including more constraints on the design variables. Here, the movement of the crane as explained in Section 3.4 should be used to describe the positions of the heel and top for different hoist radii.

The above recommendations would mostly influence the constraints, but also the objective functions might be altered. Instead of only minimizing the structural weight, other factors such as costs or overturning moment might be added to the objective function. This would be an easy alteration for Tetrahedron if they consider it necessary in the future. For this research, using the same objective function made the comparison of the different test cases, methods and the existing algorithm the easiest.

Recommendations - Optimization methods

Since MMA is the best method for the crane optimization problem, it might be worth it to improve this method even more. The research on MMA also suggests this, as Svanberg extended the MMA method to a Globally-Convergent Method of Moving Asymptotes (GCMMA) [3]. This uses inner and outer iterations, where only for each outer iteration, the gradients of the constraints and objective function have to be calculated. This reduces the computation time. Next to that, the subproblems are strictly

convex, which means that the subproblems have a unique optimal solution. According to the test case that is considered in [3], the GCMMA method needs less iterations and is less sensitive to the method's parameters and therefore more robust.

The Python code is now written as a separate module, but for an easy use for all employees of Tetrahedron it has to be implemented as a part of ACE. Using files from the ACE code, some efforts have already been made to write the code in a similar matter. Next to that, for the documentation of the code, instruction videos are made that explain all the files separately, together with a flowchart that gives an overview on the functions of all the files. With this, the Tetrahedron employees should be able to use the optimization module by themselves.

Bibliography

- [1] I. C. Vooijs, "Literature report; Shape and size optimization of an offshore crane," tech. rep., Delft University of Technology, Delft, Netherlands, March 2020.
- [2] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, p. 877–900, 1999.
- [3] K. Svanberg, "MMA and GCMMA, versions September 2007," report, Optimization and Systems Theory, KTH, Stockholm, Sweden, 2007.
- [4] D. Kraft, "A software package for sequential quadratic programming," *Forschungsbericht-Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- [5] L. v. Heel, "Maasvlakte krijgt 's werelds grootste windmolen op land." www.ad.nl/rotterdam/maasvlakte-krijgt-s-werelds-grootste-windmolen-op-land~a12796b9/, 2018. Online; accessed 17 October 2020.
- [6] Tetrahedron, "About a simple innovative crane design for ultra high offshore wind turbine lifting." Internal presentation, January 2019.
- [7] M. P. Bendsøe and O. Sigmund, *Topology optimization: theory, methods, and applications*. Berlin: Springer, 2019.
- [8] T. Dede and Y. Ayvaz, "Combined size and shape optimization of structures with a new meta-heuristic algorithm," *Applied Soft Computing Journal*, vol. 28, pp. 250–258, 2015.
- [9] N. Petrović, N. Kostić, and N. Marjanović, "Comparison of approaches to 10-bar truss structural optimization with included buckling constraints," *Applied Engineering Letters*, vol. 2, no. 3, pp. 98–103, 2017.
- [10] W. Hare, J. Nutini, and S. Tesfamariam, "A survey of non-gradient optimization methods in structural engineering," *Advances in Engineering Software*, vol. 59, pp. 19–28, 2013.
- [11] V. N. Jamariya, D. D. Panchal, and S. R. Tare, "Structural optimization of truss using finite element analysis," in *2018 International Conference on Smart City and Emerging Technology, ICSCET*, 2018.
- [12] K. Svanberg, "The method of moving asymptotes - a new method for structural optimization," *International Journal for Numerical Methods in Engineering*, vol. 24, no. June 1986, pp. 359–373, 1987.
- [13] R. C. Hibbeler and K. B. Yap, *Mechanics for engineers: Statics*. Singapore: Pearson, 2013.
- [14] R. C. Hibbeler, *Mechanics of materials: Eighth edition in SI units*. Harlow, England: Pearson, 2018.
- [15] C. A. Felippa, "Introduction to Finite element methods." Lecture notes from University of Colorado, Aerospace Engineering Sciences Department, 2000.
- [16] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 2 ed., 2006.
- [17] L. F. F. Miguel and L. F. Fadel Miguel, "Shape and size optimization of truss structures considering dynamic constraints through modern meta-heuristic algorithms," *Expert Systems with Applications*, vol. 39, pp. 9458–9467, aug 2012.

- [18] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Information Sciences*, vol. 183, pp. 1 – 15, 2012.
- [19] X. S. Yang, *Nature-inspired metaheuristic algorithms*. Frome, United Kingdom: Luniver press, 2008.
- [20] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, pp. 60 – 68, 2016.
- [21] R. T. Haftka, Z. Gürdal, and M. P. Kamat, *Elements of Structural Optimization. Third revised and expanded edition*, vol. 11. Springer Netherlands, 3 ed., 1992.
- [22] K. W., "Minima of functions of several variables with inequalities as side constraints," Master's thesis, University of Chicago, 1939.
- [23] C. Fleury, "Shape optimal design by the convex linearization method," in *The optimum shape: automated structural design* (B. J.A. and B. M.E., eds.), General Motors research laboratories symposia series, Springer, Boston, MA, 1986.
- [24] J. H. Starnes and R. T. Haftka, "Preliminary design of composite wings for buckling, stress and displacement constraints," *J. Aircraft*, vol. 16, pp. 564 – 570, 1979.
- [25] C. Fleury, "CONLIN: an efficient dual optimizer based on convex approximation concepts," *Structural optimization*, vol. 1, pp. 81 – 89, 1989.
- [26] C. Fleury, "First and second order convex approximation strategies in structural optimization," *Structural Optimization*, vol. 1, no. 1, p. 3–10, 1989.
- [27] K. U. Bletzinger, "Explicit approximation of equality constraints," *Optimization of structural systems*, vol. 2, pp. 555–567, 1993.
- [28] K. Schittkowski and C. Zillober, "Nonlinear programming: algorithms, software and applications. from small to very large scale optimization," in *System Modeling and optimization* (J. Cagnol and J. Zolesio, eds.), ch. 5, pp. 73 –107, New York: Kluwer Academic Publishers, 2003.
- [29] K. Schittkowski, C. Zillober, and R. Zotemantel, "Numerical comparison of nonlinear programming algorithms for structural optimization," *Structural optimization*, vol. 7, pp. 1–19, 1994.
- [30] D. Wang, W. H. Zhang, and J. S. Jiang, "Combined shape and sizing optimization of truss structures," *Computational Mechanics*, vol. 29, no. 4-5, pp. 307–312, 2002.
- [31] Y. Chen, A. Bhaskar, and A. Keane, "A parallel nodal-based evolutionary structural optimization algorithm," *Structural and Multidisciplinary Optimization*, vol. 23, no. 3, pp. 241–251, 2002.