# Efficient constraint multi-objective optimization with applications in ship design

Winter, R. de

## Citation

Winter, R. de. (2024, October 8). *Efficient constraint multi-objective optimization with applications in ship design*. Retrieved from https://hdl.handle.net/1887/4094606

# Chapter 3

# Ship Design Problem Characteristics

Ships are design intensive and belong to the most complex engineering objects worldwide [35, 106]. In this chapter, the aim is to detect patterns in this complex engineering problems by answering research question 1: *What are typical ship design optimization problem characteristics?* This chapter therefore describes ship design problem characteristics, their evaluation methods, guidelines for holistic ship design optimization, and an illustrative example.

## 3.1 Introduction

Traditionally, initial ship designs are generated by modifying existing designs by experienced architects [113]. The decisions made early in the design process are made using limited time and budgets and are difficult and costly to reverse in later design stages when these decisions turn out to be sub-optimal [68, 115]. In this traditional way, ships are designed and optimized in an iterative design process illustrated by the design spiral [58]. In the design spiral as presented in Figure 3.1 several sequential decisions are made in such a manner that the objectives are optimized and constraints imposed by physics and regulating authorities are met. However, this approach is time-consuming and the design process likely leads to a so-called locally optimal solution. The locally optimal solution is found due to the fact that it is impossible for human experts to consider all the dependencies between the decision variables, the constraints, and the objectives [112].

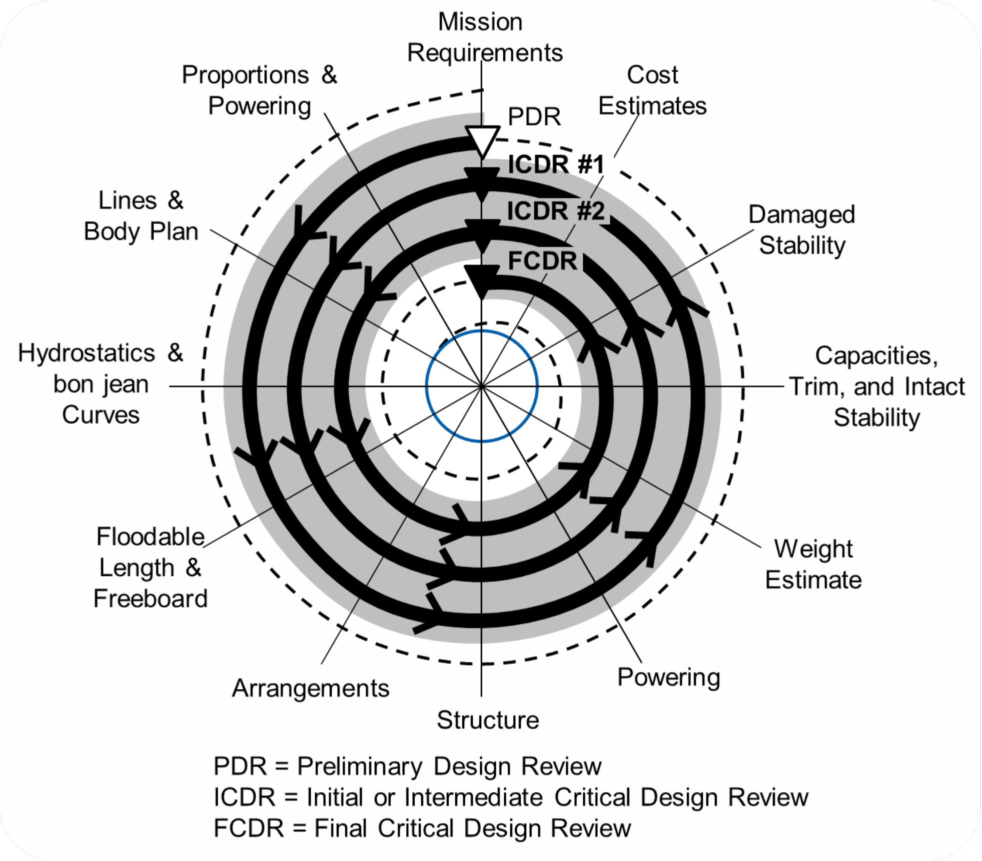

**Figure 3.1:** Classic Design Spiral [58] where iteratively details are added and where iteratively the design is adjusted by checking all ship design aspects requirements and optimizing the objectives one by one for various ship design stages.

In practice, this means that a ship is optimized not in one iteration but by designing and building multiple ones. This is because in the first design phase, the main dimensions of a ship are to be decided on. This brings many complex engineering questions which will have a very high impact on the final result. However, it is these decisions that need to be made in the shortest amount of engineering time. The more unique the ship is, the higher the challenge will be to answer the questions with confidence. Answering these complex engineering questions is usually done based on experience and heuristics. As the design progresses, these early choices are hard to reverse without major delay. This pushes the designers towards risk mitigation instead of innovation [149]. Innovation is therefore often seen over several designs, each being a small improvement over the other.

Decisions in the ship design process can be made using two different design philosophies, the empirical and the simulated design method. The empirical design method is based on reference data of similar built vessels. The simulated design method uses estimations, calculations, and simulations to optimize the economical and physical characteristics of the to-be-designed vessel.

### 3.1.1 Empirical Design Method

In the Empirical Design method, the main dimensions are based on similar-built vessels. A similar vessel is marginally improved or the data from a set of similar vessels is used to make a regression model, after which empirical design formulas can be deducted. Examples of empirical design equations to estimate light ship weight have been created by Watson [45], Schneekluth and Bertram [129], D'Almeida [55], Andrews [7], Molland [108] and Papanikolaou [114]. With these equations, a naval architect can easily estimate ship design parameters and their corresponding KPIs. The equations are usually calibrated for different ship types. It is however important to handle the relations carefully. It is the naval architects' job to update the relationships whenever possible [108]. In reality updating is often not frequently done and the equations are only available for the most popular ship types. A second note to keep in mind is that extrapolation of the regression models remains problematic [114].

### 3.1.2 Simulated Design Method

If no or very little data of similar ships is available, or if the ship type is uncommon so there are no existing empirical formulas, it is challenging to use the empirical design method. The naval architect in this case is forced to design a ship from scratch using estimations, calculations, and simulations. An efficient way to do this is by utilizing a parametric 3-D model connected to simulation software. General design knowledge and design experience are used to set up this 3-D parametric model. Examples of the parametric modeling approach can be found in e.g. [100, 121, 149]. Typically, after a parametric model is set up, the designs are optimized for their economical and physical characteristics with optimization algorithms [149].

A second, more automated, parametric design method has recently been proposed by Charisi et al [37]. In this work, it is shown that knowledge-based engineering is a good option when designing a ship when not enough similar ship data is available.

With knowledge-based engineering, general multidisciplinary knowledge is translated into individual product models/building blocks that represent a small part of a vessel. These product models are then used, scaled, and combined into an entire ship design in an object-oriented way.

### 3.1.3 Radical Design Choices

In order to make more radical design changes with the empirical or simulated design method in a short time and to manage the involved risk, a holistic ship design method is needed to assist the naval architect in making informed decisions. Ship designs can be optimized much more efficiently in a holistic manner where all ship design aspects are considered simultaneously. In recent years, parametric models have been used in combination with optimization algorithms to explore the design space early in the design process, enabling engineers to gain better insight into the design problem, reducing the probability for design changes later, and increasing the chances for optimal cost and operational performance [86, 149].

## 3.2 Holistic Ship Design Optimization

When optimizing a ship all disciplines should work together to come to an optimal solution. This solution however is often hard to find since there are many choices to be made. As mentioned earlier, in practice, they are made based on experience and heuristics while they should be made based on estimated performance data. Fortunately, holistic optimization is slowly being adopted in the design industry.

Holistic optimization has been applied in different fields of engineering. For example, it has been studied before in ship design optimization by Papanikolau et al. [113]. It has also been used to optimize aircraft designs [5] and passenger cars [132]. Similar techniques have also been used to optimize plant-wide production processes [36]. All these optimization problems have in common that they are solved by taking a holistic, integrated perspective while solving the problem with an optimization algorithm.

The holistic ship design approach from this work consists of several elements, namely:

1. Different software packages in which the vessel's Key Performance Indicators (KPIs) are evaluated.
2. An optimization problem definition with a parameterized design, and objective and constraint functions.

3. An optimization framework where the optimization algorithm, the problem definition, and the different software packages are coupled.

Formulating a ship design problem is however not always straightforward, therefore several guidelines are defined.

### 3.2.1 Ship Design and Evaluation Software

There are various maritime software packages in which ship design aspects can be defined and KPIs can be evaluated. The C-Job Design Circle [149] in Figure 3.2 illustrates this.

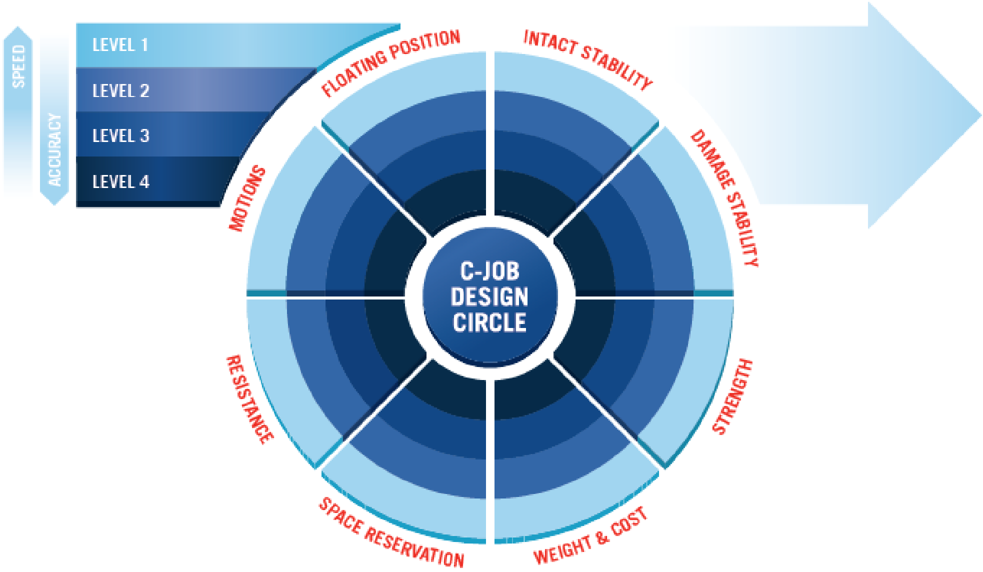

**Figure 3.2:** The Accelerated Concept Design Circle illustrates the method where all KPIs are considered in every iteration. The level of accuracy of the design simulators can be picked and mixed by the user. High-fidelity simulations that deliver highly accurate estimations require more computational effort compared to low-fidelity approximations which are therefore way faster.

In the design circle, the following design aspects are highlighted: floating position, intact stability, damage stability, strength, weight & cost, space reservation, resistance, and motions. As can be seen in the design circle (Figure 3.2), there are multiple levels of accuracy per design aspect. A high level (level 4) has a higher accuracy and requires more computational effort compared to a low level (level 1) that is less accurate and is computationally cheaper. As an example, level 1 to level 4 are given for resistance at service speed (also sometimes referred to as power requirement):

LEVEL 1 Is based on empirical design methods, is very coarse, and typically consists of a regression line that is used to estimate the resistance. This regression line is fitted on a dataset from a set of existing vessels that have a similar desired operational speed and main dimensions as the vessel to be designed. This regression line is then used to estimate the required power of the new to-be-designed ship. Estimation with this method requires almost no computational effort but is also very coarse since minor changes in the design (e.g. a change in bulb size) do not reflect in any change in estimated resistance. This method can however be a good guideline for helping to choose the main dimensions of the vessel. This method for required power estimation in combination with verifying if the selected set of vessels is really operating at the design speed can be accurate in cases where the to-be-designed vessel is very similar to the existing vessels.

LEVEL 2 The empirical Holtrop & Mennen method [79] is used to estimate the power requirement. The Holtrop & Mennen method is a well-known resistance estimation method. It considers factors like hull shape, dimensions, and speed to calculate different resistance components, including residuary, wave-making, viscous, and appendage resistance. Making estimations with this method however requires a bit more effort compared to level 1 because the Holltrop & Mennen method requires more information about the hull shape (Block Coefficient, Longitudinal center of buoyancy, Prismatic Coefficient, bulb size, U versus V shape frames), dimensions and speed. The downside of this method is that it expects the hull to be well-faired and hydrodynamically optimized, therefore the results are for an optimized and well-faired hull. However, often in the initial design stages where the Holtrop & Mennen estimation method is used, the hull is not yet optimized and faired. The Holtrop & Mennen estimated resistance therefore does not reflect the resistance of the current hull but what the resistance is after the hull is optimized and faired. Furthermore, the method is not suited for vessels that are operating at very high speeds, or have an odd length-over-breadth ratio and are therefore very different than the vessels in the database used to develop the Holtrop & Mennen method.

LEVEL 3 A potential flow solver like the one from Tahara et al [140] or Rapid from Raven [122] can be used for hull resistance and drag estimation. A typical potential flow solver is a computational tool used to analyse how water flows around a ship's hull. It simplifies the Navier-strokes equations by omitting viscosity (it does not take vortices, friction, or flow separation into account) and assuming incompressible and irrational flow. The results from a potential flow calculation consist of the wave-making resistance, the dynamic trim and sinkage, the wave pattern, pressure distribution, and the flow lines on the hull. Therefore, potential flow solvers could be used to optimize the bow of the ship but are less feasible for optimizing the aft ship where the thickness of the boundary layer is significant and flow separation and viscous forces are important. Potential flow solvers can often excellently be used in initial hull shape design stages. A simple potential flow calculation typically requires a few minutes.

LEVEL 4: A Reynolds Averaged Navier-Stokes Equation (RANSE) analysis, often called Computational Fluid Dynamics (CFD) analysis, is used to analyze the fluid flow behavior around the hull. This is the most accurate level of the design circle and requires the most computational effort. Typically CFD analysis is done with commercial software like STAR-CCM+ [135] or open source software like OpenFOAM [168]. The results from a RANSE analysis are typically pressure drag, friction drag, dynamic trim, and dynamic sinkage, the pressure on the hull, the shear stress on the hull, wave pattern, and flow lines on the hull. The same results (with similar accuracy) can be obtained by executing model tests in a basin. It is typically used to optimize the hull in one of the final stages of the design where the final details are to be sorted out about the fairing, and the ideal appendage location and size. A simple resistance calculation in calm water typically requires 20 minutes on a high-performance computer with 144 cores. Very detailed full-scale ship simulations of a complete model in realistic operating conditions and turbulent flow would require much more computational resources.

Besides the resistance and power requirement for ship design, the other design aspects such as stability, strength, weight, motions, floating positions, and space reservations can also be defined and computed at different levels of accuracy. The speed and consequently the level of accuracy at which the design variants are evaluated can be picked and mixed by the user. This means that if resistance is considered very important a RANSE calculation can be selected, while strength for example can be calculated with a simple longitudinal bending moments check, and if for example the space reservation is guaranteed to be feasible by the parametric model, then evaluation for this KPI can be completely skipped. The level of accuracy and duration of the evaluation influences how many evaluations can be done in the often short available time in the early design stages. A simple volume check of a cube shape tank can be executed thousands of times in a very short amount of time while a RANSE analysis is more expensive in terms of computational cost and can therefore be computed much less often.

The diverse nature of ship design tasks leads to a wide variety of specialized software packages. Noteworthy commercial software packages used in this work are:

RHINOCEROS 3D or Rhino in short, is primarily used in ship design to create 3D models of vessels. This allows designers to shape hulls, parameterize hulls, fair hulls, and define tanks and rooms. The programming, import, and export functionality of Rhino makes it convenient to integrate Rhino with other analysis tools for structural and hydrodynamic assessments.

NAPA software is used to model, parameterize, and evaluate any type of vessel. The NAPA software is mainly used to define the space reservation with room and tank arrangements and evaluate the damage and intact stability criteria of vessel designs. Other aspects that can be calculated with NAPA are seakeeping, floating position, and structural aspects for weight calculations. The programming capabilities in NAPA make the software modular, easy to adjust, and give users the possibility to establish a connection to other software packages.

DELFTSHIP maritime software, is a visual hull modeling and stability analysis program that allows. If modeled correctly, all the data required for damage and intact stability can be calculated together with the bending moments and shear forces. Just like in other design software packages, it is possible to program custom scripts in DELFTship so that it can be coupled to other packages.

STAR-CCM+ is powerful computation fluid dynamics (CFD) software package developed by Siemens. It is mainly used for hull hydrodynamics, to assess the propulsion requirement for the vessels. These calculations are crucial when optimizing hull shapes for fuel consumption reduction.

All these software packages require commercial licenses and if multiple simulations are to be run in parallel, multiple licenses are required during the entire evaluation time. The commercial license requirement together with available computing resources typically drastically limits the number of allowed evaluations during the optimization process. Therefore, it is important to pay attention to how many evaluations are possible in the available time and how the number of evaluations can be decreased.

### 3.2.2 Optimization Problem Definition Guidelines

There are some guidelines to follow when setting up a ship design optimization problem (or any optimization problem for that matter) in a holistic manner. The guidelines deal with the parameterization of the solution, and how to formulate the constraint and objective functions.

#### Parameterization Guidelines

In the parameterization stage of ship design, the naval architect can use imagination and creativity to create an innovative and new design that should be optimized. Important to remember when parameterizing and later optimizing a ship design model is that the best vessel that can be found by the optimization algorithm is dependent on the parameterization of the vessel. So if the parameterization is bad, the results will be bad. To give two illustrative examples:

1. If the goal is to make an as light as possible vessel, but the parameterization doesn't allow for vessels that are smaller than the design that is parameterized, it is very unlikely that a lighter vessel can be found with this parameterization.
2. If one wants a design with as small as possible hull resistance at design speed, parameterizing the interior will probably only help a tiny bit in reducing the resistance because it has a marginal influence on the floating position. It would be much better in this case to parameterize the hull.

Therefore, the ship should be parameterized in such a manner that the performance of the vessel also depends on the decision parameters that are defined in the parameterized model. Defining decision variables that only marginally influence the constraint and/or the objective values will only have marginal influence so they can better be neglected or be regarded as a constant throughout the whole process.

Secondly, it is important that after all the decision variables have been set, the geometry is formed accordingly. After this, the geometry should not be changed anymore (for example to meet a constraint) until a new combination of decision variables is chosen. This way, the input truly corresponds with one unique geometry. After the geometry has been evaluated, it will be these decision parameters that define the geometry and the corresponding constraint and objective values. This approach can sometimes result in infeasible solutions since some combinations of decision parameters will result in a geometry that violates one or more of the constraints. This is not a major problem since optimization algorithms directly learn the relationship between the decision parameters of the solutions and the constraint and objective scores and therefore will learn to propose better combinations in the next iterations. Because the relationship between the parameters and the constraint and objective functions is learned by the optimization algorithms, the number of parameters can also have a significant impact on how many function evaluations are required to find (a set of) optimal solution(s).

### Constraint Function Guidelines

A constraint function is there to check if the solutions comply with the hard boundary conditions set by the stakeholders. A guideline to take into consideration when defining constraint functions is to make sure that the output of the constraint function is real-valued and preferably continuous. Bad practice is a simple Boolean constraint value corresponding to the constraint being violated or met. A real-valued constraint function is important since optimization algorithms can learn by how much the constraints are violated or met. This information can be learned by the optimization algorithm and can be crucial since, more often than not, the optimal design configuration is located on the border of the feasible area. The constraints are defined as function inequalities: $g_i(\mathbf{x}) \leq 0$ where $\mathbf{x}$ are the decision parameters and $g_i()$ is one of the evaluation functions used to evaluate one of the $m$ constraints. Note that here 0 is the feasibility boundary, exactly how it should be formulated according to our problem formulation in Equation 2.1. However, in reality, a typical constraint function $g'(\mathbf{x})$ consists of three parts: a continuous function $g'$ which takes input arguments $\mathbf{x}$ that represent the decision variables that can modify the solutions, and a constraint boundary $c$ that represents a maximum or minimum value for the constraint. These three parts taken together form for example the inequality constraint $g'(\mathbf{x}) - c \leq 0$. This inequality constraint can then be rewritten to its general form: $g(\mathbf{x}) \leq 0$. When constraints have a minimum value they can be rewritten without loss of generality to a function with a maximum value by multiplying by $-1$. Note that the function $g'(\mathbf{x})$ does not need to be a function that can easily be written by mathematical formulas. The function $g'(\mathbf{x})$ can also be a software program that calculates any of the KPIs.

In some design cases, also contain hard equality constraints. Sometimes, they can simply be avoided by adjusting the parameterization of the design, while in other cases an equality constraint is inevitable. In these cases there is an equality constraint $h(\mathbf{x}) = 0$, it should be replaced with two inequality constraints. This way, there is no need to resort to special equality constraint handling methods. Good practice is however to add a small margin ($\epsilon$) around the constraint boundary so that in case there are tiny numerical instabilities this does not influence the feasibility of the solutions.

$$h(\mathbf{x}) = \begin{cases} g(\mathbf{x}) - 0.5\epsilon \leq 0 \\ g(\mathbf{x}) + 0.5\epsilon \geq 0 \end{cases} \tag{3.1}$$

#### Objective Function Guidelines

The objectives are measurable KPIs that the stakeholders look to improve as much as possible, it can be anything as long as it is numerically measurable and dependent on the decision parameters. When defining the objective functions, one should define a real valued, and again preferably continuous, function that only represents one objective at a time. This is preferred above a weighted sum of all objectives. Similarly, one should not add penalties to the objective score if the design violates any of the constraints. This way, the objective function remains smooth and the true optimal solution can be found more easily by optimization algorithms. Objectives are typically conflicting and non-commensurable. Therefore, often one objective can not be improved without sacrificing another objective. As a consequence, there is not one perfect solution but a set of non-dominated solutions that form the Pareto optimal set. The definition of the Pareto optimal set is used as described in 2.9.

### 3.2.3 Optimization Framework

The different software packages, the formulation of the objectives, constraints, and parameterization come together in the holistic Accelerated Concept Design (ACD) framework [149]. Because every aspect of the ACD framework is fully automated, the designs can easily be varied and optimized towards, for example, the total cost of ownership. By using any of the surrogate-assisted multi-objective optimization algorithms from Chapter 5, the whole design space can be considered which makes it more likely that a globally optimal solution will be found.

## 3.3 Illustrative Ship Design Optimization Problem

In this section, a example ship design optimization case is presented for illustration purposes. In practice, every ship design optimization problem is different but in theory, it is nothing more than defining a ship design optimization problem and solve it. Note that the design and parameterization stage is where the naval architect can use their imagination and creativity to come up with new innovative designs.

## 3.3. Illustrative Ship Design Optimization Problem

To illustrate this process of defining and optimizing a vessel, a Trailing Suction Hopper Dredger (TSHD) designed by C-Job Naval Architects is described and optimized. This TSHD will be optimized to reduce total cost of ownership. Total cost of ownership consists of Captial Expenses (CapEx) and Operational Expenses (OpEx). The original design with the annotated decision parameters is shown in Figure 3.3.

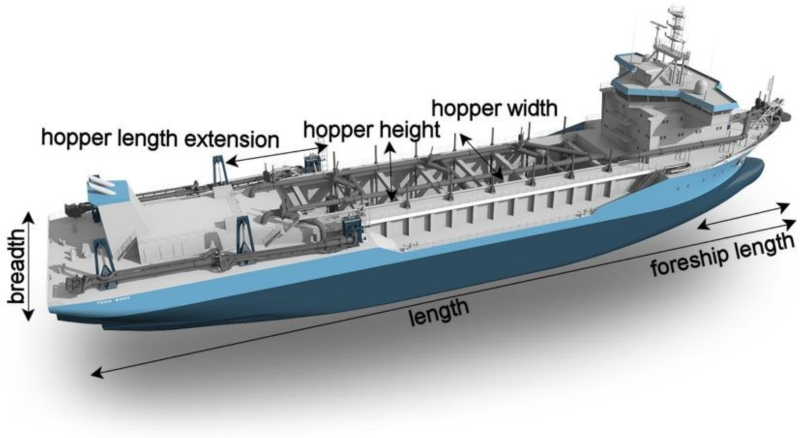

**Figure 3.3:** Trailing Suction Hopper Dredger with design parameters annotated.

### 3.3.1 Parametric Geometry

The first step is to generate a parametric geometry of a vessel. The geometry depends on decision variables which are numerical quantities for which values can be varied in the optimization process [41]. The decision variables are Free-Form Deformation (FFD) [43] parameters to modify the hull and parameters that are used to rearrange the bulkheads in the vessel.

#### Repositioning Bulkheads

The rooms inside the hull can easily be parameterized. This is very useful since it will be the rooms and their loading that influences the floating position, intact stability, damage stability, draft, heel, trim ect. Parameterizing rooms can be done by moving locations of bulkhead along the $x$, $y$ and $z$ axis. As an example, the bulkheads that define a hopper of a TSHD are moved along the $x$-axis in Figure 3.4. This way, the hopper can be made smaller, larger, or the location of the hopper can completely be moved.

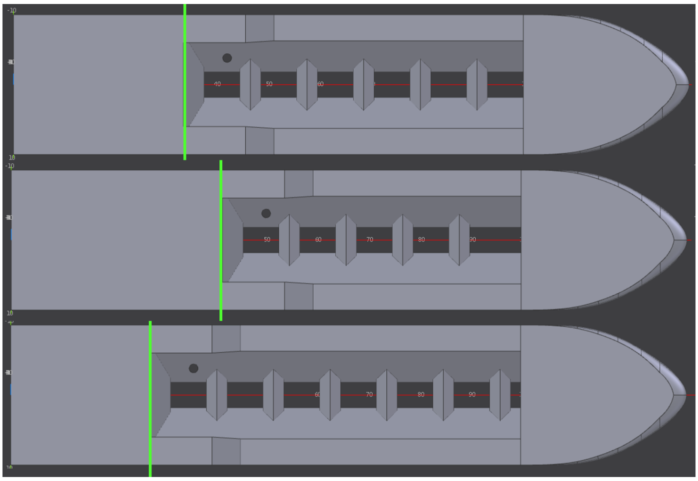

**Figure 3.4:** Bulkheads moved forward and backward to define different hopper lengths. The bulkhead positions are marked in bright green.

Good practice is to make bulkheads dependent on each other. To illustrate this the following example is given: if bulkhead 1 is located at frame 0, the room between bulkhead 1 and bulkhead 2 is 7 frames long, then bulkhead 2 is located at frame 7. The room between bulkhead 2 and bulkhead 3 is 10 frames long. Then bulkhead 3 is located at frame $7 + 10 = 17$. In case the length of room 1 is increased by $x$ frames, then move bulkhead 2 to $7 + x$ and move bulkhead 3 to $7 + x + 10$. In case the length of room 1 is decreased by $x$ frames, then move bulkhead 2 to $7 - x$ and move bulkhead 3 to $7 - x + 10$. This dependency is displayed in Figure 3.5.

### Free-Form Deformation

When using automated optimization tools for altering the hull form of the vessel, a challenge may arise with keeping the fairness (and other important characteristics) of the hull surface. Using the Free-Form Deformation technique, the hull form change undergo rather radical changes in a robust way. The optimization by FFD is done by applying a lattice (box) around the surface or part of the surface which is to undergo transformation. The lattice is then deformed by translating it's vertices. The control points of the circumscribed surface will consequently undergo transformation leading to a new surface geometry. An example of a FFD applied on the fore- and aft-ship of a hull is illustrated in Figure 3.6.

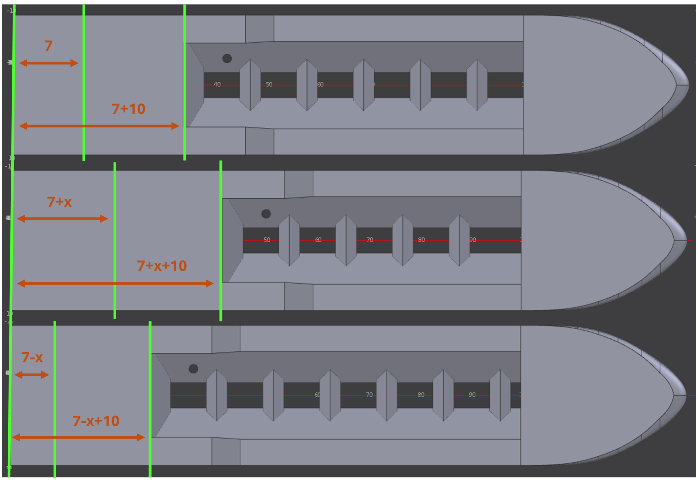

**Figure 3.5:** Bulkheads moved forward to make rooms larger or smaller. The bulkhead locations are marked in bright green and the red arrow lengths indicate the distance between the bulkheads.

### Trailing Suction Hopper Dredger Parameters

The TSHD design that is parameterized for this illustrative example is presented in Figure 3.3. The model has the following six decision variables: $\Delta_{breadth} \in [-1.6, 3.4]$, $\Delta_{length} \in [-2.8, 9.8]$, foreship length $\in [16, 22]$, hopper length extension $\in [5, 9]$, hopper breadth $\in [5, 9]$, hopper height $\in [12, 16]$. The ship length and ship breadth are annotated with a $\Delta$ because these variables are used to apply a FFD on the initial hull. The other parameters are all bulkhead locations which can be moved and/or extended along the $x$, $y$, and $z$ direction.

## 3.3.2 Ship Design Constraints

For the TSHD some quite specific constraints have to be set in order to create sensible design variations that meet the stakeholders' demands and which do not violate any of the regulations. These constraints mainly focus on the hopper volume, stability criteria, space reservation, fuel capacity, water ballast requirement, and the dredging area where the vessel will be dredging and dumping. The details of these constraints are defined below:

HOPPER VOLUME: The hopper where the dredged soil is dumped in, should at least be 8000 cubic meters to meet the client's demand. The inequality constraint for the hopper is therefore: $8000m^3 - VOL(hopper) \leq 0$.

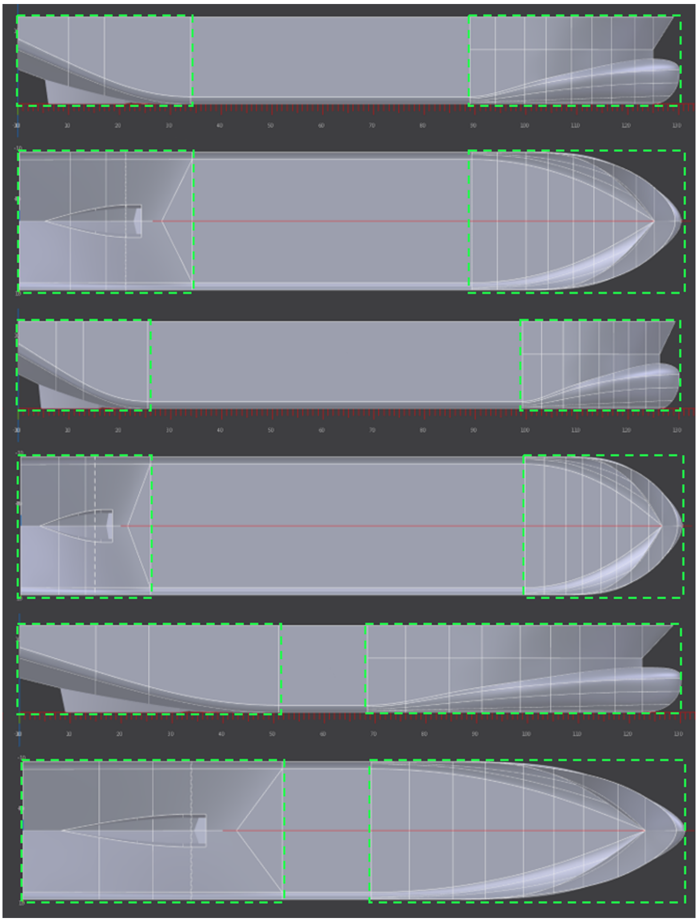

**Figure 3.6:** Free Form Deformation applied to fore- and aft-hull.

Fuel capacity: The design variation has 9200 kW installed power, the fuel type is Marine GasOil (MGO), and the design variation should at least be able to sail for 21 days. Therefore a relatively conservative fuel capacity constraint is defined: $820m^3 - VOL(fueltanks) \leq 0$.

Propulsion Engine: The engine room should at least be large enough to accommodate the required space and volume of the propulsion engine. Therefore, the propulsion engine and the engine room are modelled and intersected. The volume of the intersection of the two rooms should be at least be equal to engine volume, if this is not the case, the engine does not fit in the engine room. $VOL(Engine) - VOL(EngineRoom \cap Engine) \leq 0$.

DREDGE PUMP: The procedure for the dredge pump is the same as for the Propulsion Engine. Therefore the constraint for the dredge pump is the following: $VOL(Pump) - VOL(PumpRoom \cap Pump) \leq 0$.

ACCOMMODATION SPACE: The accommodation on the design variation should host 29 crew members. For 29 persons an estimation of the required space is made, this estimation resulted in a required 1100 cubic meters accommodation space. Therefore the inequality constraint for the accommodation is the following: $1100m^3 - VOL(Acc) \leq 0$.

DRAFT: The design variation should be able to sail in shallow waters, therefore the draft $T$ should not exceed 7 meters when fully loaded. The corresponding inequality constraint is therefore: $T - 7m \leq 0$ such that when the design variation is fully loaded it is still capable of sailing in shallow waters.

FORE PEAK BULKHEAD: The fore peak bulkhead, also referred to as collision bulkhead, should be positioned according to the International Convention for the Safety Of Life At Sea (SOLAS) [110]. The constraint from SOLAS is as follows: $\min(0.05L, 10m) \leq d \leq \max(0.08L, 0.05L + 3)$, where $L$ is ship length and $d$ is the collision bulkhead position. Rewritten and separated into two inequality constraints, this rule then gives the following inequalities: $\min(0.05L, 10m) - d \geq 0$, and $\max(0.08L, 0.05L + 3) - d \leq 0$.

STABILITY CRITERIA: The stability requirements are evaluated by expressing the applicable regulatory requirements as a meta-centric height value $GM_c$. Given any possible loading condition of the ship, the obtained meta-centric height $GM_o$ of the design variation should at least be larger or equal to the prescribed metracentric height value to pass all the stability requirements. The stability function inequality therefore is the following: $GM_o - GM_c \leq 0$.

TRIM: Given different loading conditions with the hopper empty and the hopper filled, without using water ballast, the trim varies. Because the TSHD at hand is a ballast-less design, it is necessary to keep the difference in trim within practical limitations. The trim has therefore been limited to a maximum difference of 2.1 meter. The inequality constraint for trim is therefore described as follows: $(\max(trim_{lc}) - \min(trim_{lc})) - 2.1m \leq 0$.

HEEL: The same procedure as for trim is followed for heel. The difference between the maximum and minimum heel is not allowed to be larger than 0.2°. The function inequality is therefore the following: $(\max(heel) - \min(heel)) - 0.2° \leq 0$

Strength is also a critical constraint, but because the steel weight is also optimized, it is made sure that only design variations are generated that comply with the strength regulations, therefore a strength constraint is not added. All the constraints defined in this section are evaluated with custom macros written in the NAPA basic software. Evaluating one design for all constraints requires a couple of minutes.

To show the severity of the constraints, an experiment was conducted to check how much of the design space was feasible. To get an indication 200 random design variations are generated and evaluated. 24% of the 200 design variations were feasible.

### 3.3.3 Ship Design Objectives

The objective in this illustrative example is minimizing the total cost of ownership of the dredger. The total cost of ownership of this particular dredger is calculated by the client, the ship owners, and the operators of the dredger. The main cost drivers of the dredger are therefore calculated to enable stakeholders to evaluate a given Pareto optimal solution and come to a decision on which one of the Pareto optimal solutions should be selected for consecutive design stages.

In this particular case, the total cost of ownership can be separated into building costs, operational gains, and operational expenses. Building costs are largely driven by the material cost and consequently can be directly linked to the weight of the vessel. Operational gains can be defined as ship speed and moved cargo, which are both fixed for this optimization case, and operational cost is largely driven by fuel consumption, maintenance, and crew costs. Minimizing the total cost of ownership can thus be achieved by minimizing the steel weight while also minimizing the resistance at service speed. These objectives are a classical example of conflicting ones, a long and slender design variation will have a smaller resistance factor and a higher estimated steel weight compared to a wider shorter variation of the same vessel.

The resistance of the different design variations is estimated with a potential flow solver [140]. Because of the nature of a potential flow solver, the mesh describing the hull shape has been idealized. The potential flow code cannot provide absolute resistance values but is suitable for comparing the resistance of different design variations.

The steel weight is more complex since the lightship weight, maximum bending moments, and loading conditions should be in balance with each other. Therefore, for every design variation a minimum steel weight needs to be estimated that meets the maximum bending moment requirement while keeping Equation 3.2 and Equation 3.3

in balance line in Equation 3.4:

$$\Delta_{actual} = L \cdot B \cdot T \cdot Cb \cdot \rho \tag{3.2}$$

$$\Delta_{required} = DWT + LSW \tag{3.3}$$

$$\Delta_{required} = \Delta_{actual} \tag{3.4}$$

where $\Delta$ is the actual and required displacement, $L$ is Length, $B$ is Breadth, $T$ is Draft, $Cb$ is Block-coefficient, and $\rho$ is the water density, $DWT$ is the Dead Weight, and $LSW$ is Light Ship Weight.

### 3.3.4 Optimization of Ship Design Problem

In the illustrative example, a potential solver is used to estimate the resistance, and NAPA macros are used to compute the constraints and the steelweight objective. This results in an optimization problem that requires 5 to 10 minutes per design evaluation. In other cases where RANSE analysis are done for resistance calculations these 5 to 10 minutes however can quickly grow to hours. If the goal remains to find good solutions on the Pareto frontier in a limited amount of wall clock time, special attention should be given to the selection process of the optimization algorithm. How to limit the number of required function evaluations, how to evaluate solutions in parallel, and how to deal with expensive and inexpensive functions is described in more detail in Chapter 5. The obtained results on this illustrative ship design problem are reported in Chapter 6 together with other Real World Application results.

## 3.4 Conclusions and Future Work

In this chapter ship design methodologies, parameterization, and evaluation methodologies are described that together form a typical ship design optimization problem. After identifying good parameterization practices, different evalulation methodologies, and guidelines on how to set up the optimization problems the holistic Accelerated Concept Design framework is set up. The parameterization of the design in the holistic framework is the most important part of the ship design optimization process. This is because the final designs are only as good as the parameterization allows. If there is enough flexibility in the parameterization that allows for improvements in the objectives and that effectively deals with the constraints, a new and better design can be generated. Secondly, the level of accuracy of the evaluations of the constraints and objectives is important. The accuracy (and evaluation software) significantly influences the required computational effort and determines together with the available commercial licenses available the total evaluation budget. The evaluation budget is critical since enough evaluations are required for optimization algorithms to converge. A large required evaluation budget is often problematic since in the conceptual design stage, the lead time is often limited.

For future work, more research is needed to find out what the best parameterization setup is for which design case. This will however require the joint effort of optimization experts, naval architects, and structural and hydrodynamic engineers.