Efficient constraint multi-objective optimization with applications in ship design
Winter, R. de

Chapter 6

Real World Applications

In this Chapter the last sub-question: *What is the performance of the proposed algorithms in real-world scenarios?* is answered. This is done by solving five real-world simulation-based ship design optimization problems using optimization algorithms. This is done to verify algorithm performance in real-world scenarios and with the algorithms that were readily available at the time the designs had to be optimized at C-Job Naval Architects. The first three problems are computationally expensive multi-objective problems with constraints that can be solved with the algorithms discussed in the previous chapters. The last two real-world optimization problems in this chapter required modifications to the optimization algorithm as these problems had one objective and the computational difference between the objective evaluation and the constraint evaluation was much larger compared to the other problems.

6.1 Trailing Suction Hopper Dredger

The first problem solved with the constraint multi-objective optimization algorithm is the Trailing Suction Hopper dredger as described in Chapter 3 and displayed in Figure 3.3. To repeat the optimization problem briefly, the problem is a two-objective problem where the steel weight and the resistance at operating speed should be minimized. The problem has 11 constraints concerning room and tank capacities, draft, trim, heel, the forepeak bulkhead, and intact stability criteria. All these constraints and objectives are evaluated in the commercial NAPA software.

The original design has a resistance coefficient of 1.08 and a steel weight of 2039 tonnes. This original design is parameterized and optimized by the ACD framework from Chapter 3 using 200 ship design evaluations proposed by the CEGO algorithm from Section 5.1.1. The framework is allowed to do 200 evaluations to evaluate the trailing suction hopper dredger. The reference point is fixed and set to $[5000, 2]$. By setting the reference point to $[5000, 2]$ the algorithm is limited to solutions with a smaller steel weight than 5000 tonnes,and a resistance coefficient smaller than 2. The experiment has been repeated five times independently with different initial starting points to check for consistency.

In a small first experiment, the severity of the constraints is investigated. In this experiment, 200 random design variations are generated and evaluated. 24% of these design variations turned out to be feasible.

6.1.1 Results of Trailing Suction Hopper Dredger Design Experiment

The results of the five independent runs were very similar. The hypervolume metric as used in multi-objective optimization between the reference point and the Pareto optimal set was on average 3819 and the standard deviation of this volume was 3.3. The parameter combinations of a typical run of 200 evaluated design variations are displayed in the parallel coordinate plot in Figure 6.1. The red lines represent the obtained infeasible solutions, the blue lines represent the feasible solutions, and the green lines represent the Pareto optimal solutions.

The Pareto optimal results of a typical run are presented in Figure 6.2. During this run, the CEGO algorithm in combination with the ACD framework found a set of 10 non-dominated design variations where the most interesting solution has a resistance coefficient of 0.87 and a steel weight of 1748 tonnes. Therefore, compared to the original design, the improved design has a 19% smaller resistance coefficient and 14% less steel weight.

6.1.2 Analysis of the TSHD Results

In Figure 6.3a and Figure 6.3b the original and the improved design are shown respectively. From the first observation, there is not a lot of difference, but the optimized result is 9 meters longer, and 50 centimeters less wide. Typically when a ship is longer more steel is needed to fulfill the strength requirements. But in this case, the hopper is higher which eases the imposed longitudinal bending moment on the ship. The extra strength results in less thick required steel plates and smaller profiles required to meet the longitudinal strength.

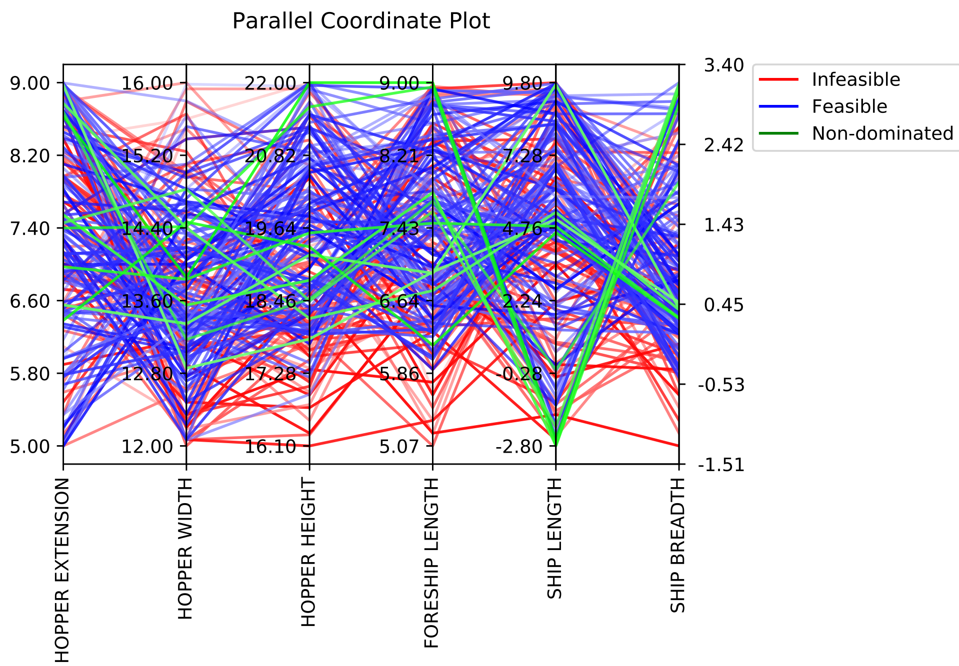


Figure 6.1: Parallel coordinate plot of the 200 different design variations. The red lines represent infeasible solutions, blue lines represent feasible solutions, green lines represent Pareto optimal solutions.

Furthermore, because the vessel is longer and less wide, the resistance also significantly decreased. This can also be seen from the wave pattern around the two design variations in the Figures 6.3a and Figure 6.3b.

6.1.3 Conclusion from Trailing Suction Hopper Dredger Study

From the results, it can be concluded that the Accelerated Concept Design framework is capable of optimizing parameterized vessels in a fully automated manner and in a very efficient way. On top of this, it is shown that this design process can reduce time and human effort while significantly improving ship designs. Furthermore, because of the use of surrogate-assisted models, and therefore objective and constrained prediction, the whole design space can be explored which would never have been an option for a human expert alone.

For future work, more software tools have to be automated and coupled to the Accelerated Concept Design framework to gather more high fidelity results. On top

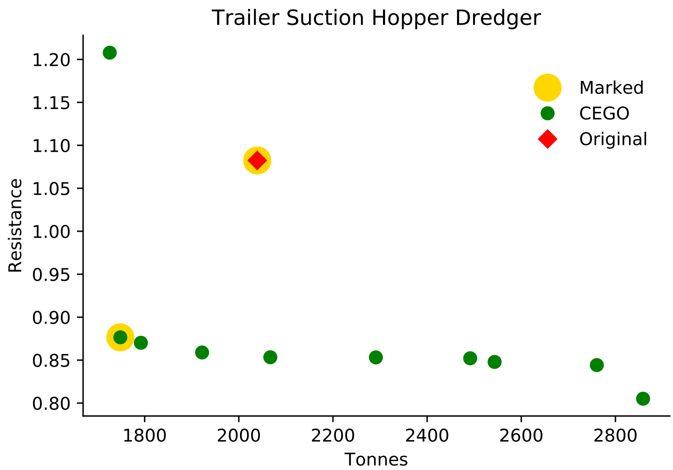
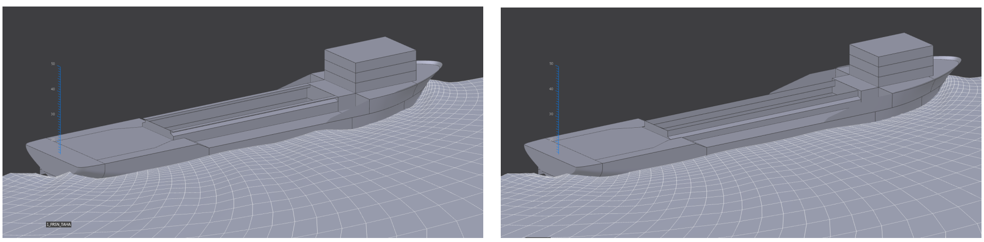


Figure 6.2: Obtained non-dominated design variations and original design. The marked solutions are the original solution and the most interesting optimized solution.



(a) Original Trailing Suction Hopper Dredger design optimized by human experts.

(b) Trailing Suction Hopper Dredger design optimized with the CEGO algorithm.

Figure 6.3: Comparison of the two Trailing Suction Hopper Dredger designs

of this, a more in-depth analysis of ship parameterization should be done to achieve even better results.

6.2 Wind Feeder Vessel

One of the advancements of the accelerated concept design framework after the TSHD optimization problem is the implementation of the Operability Robustness Index to optimize a Wind Feeder vessel. The SAMO-COBRA algorithm has been used in practice to design a wind feeder vessel to support the installation of windmills at sea. Although high winds are good for power production, they usually also result in rough seas. These rough seas around the wind park installation sites increase the demand for reliable vessels. The impression of such a wind feeder vessel is presented in Figure 6.4. This vessel has been designed and later optimized at C-Job Naval Architects [30]. This vessel is specifically designed to support the construction of wind farms and to transport the materials from the shore to the installation sites for the US market.

Figure 6.4: Impression of the Wind Feeder Vessel design by C-Job Naval Architects.

The objectives of the optimization case of the wind feeder vessel are to have a robust seakeeping performance to maximize the year-round operability, while also keeping the operational cost and capital expenses at a minimum. The operability can be optimized by maximizing the so-called Operability Robustness Index (ORI)[71]. The ORI objective takes the area of operation into account and therefore can be optimized for a certain wave spectrum. In this case the Pierson Moskowitz spectrum is used as recommended by the DNV-GL maritime classification bureau [52]. The seakeeping assessment is done with a strip theory code of NAPA[1]. Strip theory is proven to be fast and reliable with sufficient accuracy for conventional hull forms [16, 69]. The capital expenses can be translated into the cost of steel that is required to build the vessel, this is roughly equal to the Lightship Weight (LSW) of the vessel. The LSW is calculated by summing the weight of all the equipment plus the minimum amount of steel that is required to fulfill the longitudinal strength requirements. The operational

[1] Intelligent solutions for the maritime industry, https://www.napa.fi/

expenses can be dealt with by minimizing the ship resistance in the water at service speed (Rt[kN]). This resistance is calculated with the Holtrop & Mennen method [79].

More details about the wind feeder design study can be found in the paper and master thesis from Bronkhorst et al. [27, 28].

All objectives, practical constraints, relevant rules, regulations, and loading conditions can be evaluated with the modular Accelerated Concept Design framework as described in Chapter 3. In this software, a parametric 3D model of the ship is set up by a naval architect after which automated software tools can evaluate any design variation in one function call. In the wind feeder vessel case, five design parameters are defined: *Aftship Length*, *Midship Length*, *Foreship length*, *Beam at Waterline*, and *Draught*. Since sea-keeping and longitudinal strength are already captured in the objectives, only two constraints are needed. The two constraints are for space reservation of the wind turbine blades and the meta-centric height for intact stability of the vessel.

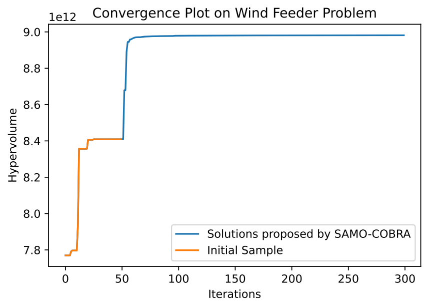
SAMO-COBRA from Section 5.1.2 is then used to optimize this ship design optimization problem. To enhance the exploration in this case study, SAMO-COBRA started with more than the advised 50 initial Halton samples. After evaluation of the initial sample, the SAMO-COBRA algorithm with the PHV infill criterion is used to propose 250 more solutions. On a desktop with an Intel Xeon Processor E3-1245 V3 quad-core processor with 16 GB of working memory, the 300 evaluations required three and a half hours of wall clock time.

6.2.1 Results of Wind Feeder Design Experiment

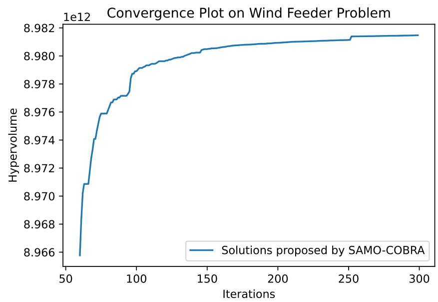
Based on the first 50 Halton samples, 36% of the design space is estimated to be feasible. Out of the total 300 design variants SAMO-COBRA was able to find 154 non dominated solutions, 35 feasible but dominated solutions, and proposed 111 infeasible solutions.

The convergence of the SAMO-COBRA optimization run is visualized in Figure 6.5a. This plot shows that after the Design of Experiments, SAMO-COBRA quickly finds the most promising solutions. When zooming in on the solutions after evaluation 60 (see Figure 6.5b) it can also be observed that the algorithm continues finding solutions that contribute hypervolume.

The Pareto frontier found on this problem is plotted in Figure 6.6. In this plot, it can be observed that the solutions on the Pareto frontier are nicely spread.



(a) Convergence Plot on Wind Feeder Problem.



(b) Convergence Plot on Wind Feeder Problem zoomed in on the last 240 iterations.

Figure 6.5: Wind feeder optimization process convergence plot (a) and zoomed in part (b).

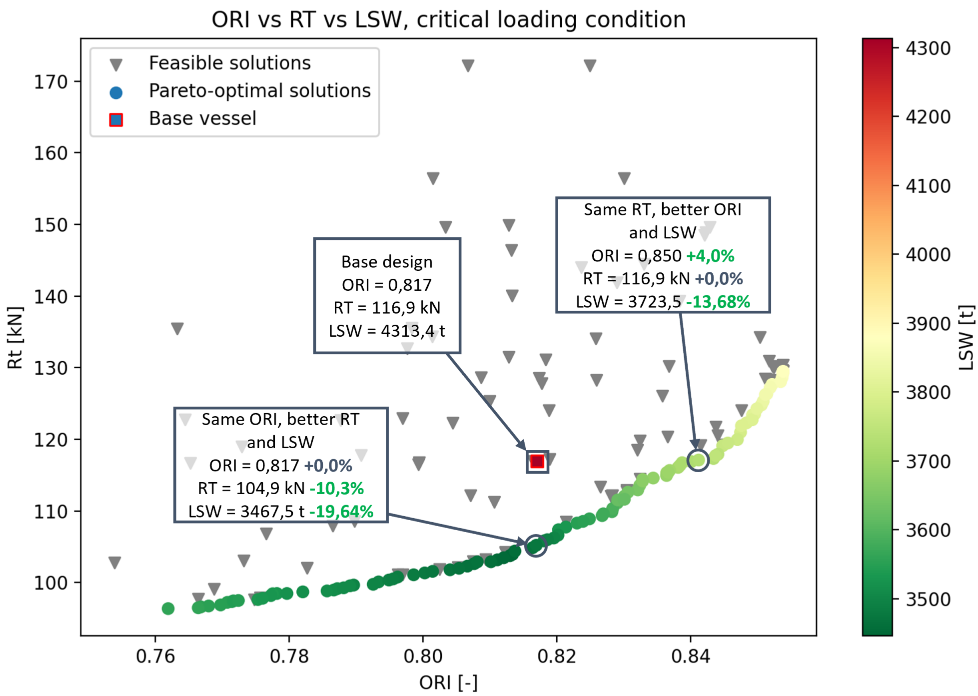


Figure 6.6: Pareto Frontier of Ship Design case with Original Design by human expert represented by a square. Objectives are maximize the Operability Robustness Index (ORI[-]), minimize ship resistance (Rt [kN]), and minimize Lightship Weight (LSW[t]).

6.2.2 Analysis of Wind Feeder Optimization Results

All evaluated, feasible solutions are visualized on the Pareto frontier in Figure 6.6. After analysing the results from the optimization study, the base design by the naval architect was shown to be much too large, causing the ship to be too heavy with a sub-optimal performance. When a few of the Pareto-optimal solutions are compared to the original, then the solution with the same ORI score has a 10.3% smaller resistance value, and 19.64% less light ship weight. The solution with the same resistance score has a 4% better ORI score, and 13.68% less light ship weight. The lightship weight reduction can be explained with that a significant reduction in length was possible.

6.2.3 Conclusion for Wind Feeder Vessel

SAMO-COBRA has been used in practice on a wind feeder optimization problem with three objectives, two constraints, and five decision variables. In this application, the algorithm demonstrates its ability to outperform the human expert in all objectives simultaneously mainly due to the fact that the original design was too large and could have been designed much smaller. The larger design did not contribute to a higher operability robustness index in the area of interest nor was it good for steel weight and resistance. However, a significant amount of wall-clock time would have been able to be saved if at the time of experimenting the IOC-SAMO-COBRA algorithm would have been available since a few of the constraints and the objectives are computationally inexpensive to evaluate.

6.3 Single Hold Cargo Ship Damage Stability Optimization

As a real-world application, the mid-ship section of a single-hold general cargo vessel design[2] as presented in Figure 6.7 is optimized for two conflicting objectives: stability ($\uparrow$ max) after potential damage (survivability), and cargo hold capacity ($\uparrow$ max).

Besides the conflicting objectives, the problem has three volumetric constraints and one regulatory constraint:

- Volumetric: The two fuel tanks should be of sufficient size so that enough fuel can be stored and the technical space should be large enough to host the equipment.

[2]Figure courtesy of C-Job Naval Architects, Hoofddorp, Netherlands.

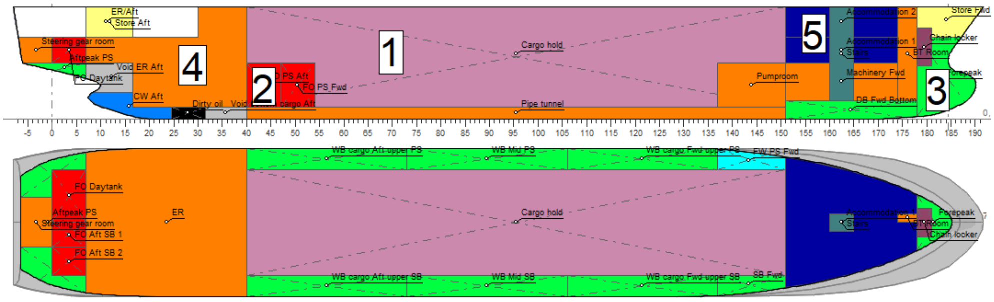


Figure 6.7: Longitudinal section and top view of cargo vessel design optimized for damage stability (survivability) and cargo hold capacity (both to be maximized). The pink compartments (1) annotates the cargo hold, the red compartments (2) are fuel tanks, the green compartments (3) are water ballast tanks, the orange compartments (4) are the technical spaces, and the blue compartments (5) are part of the crew accommodation.

- Regulatory: The attained damage stability index (survivability) score should be larger compared to the required damage stability index.

The objectives and constraints depend on 17 geometric parameters, which influence the longitudinal and transversal positioning of the bulkheads and the heights of openings. The bulkheads split the different compartments and tanks together with the height of decks and openings in the vicinity of the cargo hold.

The evaluation of the damage stability (survivability) objective and the corresponding comparison between the required damage stability constraint is computationally expensive. Evaluation of the damage stability index requires a run of the commercial maritime simulator Delftship pro[3]. The volumetric objective and the three volumetric constraints are inexpensive to evaluate. This offers the opportunity to optimize the design problem with the IOC-SAMO-COBRA algorithm from Section 5.3. The inexpensive constraints and objective are directly used in the IOC-SAMO-COBRA algorithm while for the expensive objective and constraint, RBF surrogates are updated and selected every iteration. More details about the ship design problem are given in [103, 102, 146].

This real-world problem is optimized in three different ways:

1. IOC-SAMO-COBRA with number of candidate solutions per iteration $p = 1$ and 300 function evaluations.
2. IOC-SAMO-COBRA with number of candidate solutions per iteration $p = 3$ and

[3]Version 14.20.343; see Delftship: Visual hull modeling and stability analysis. `https://www.delftship.net/`

501 function evaluations.

3. SA-NSGA-II with number of candidate solutions per iteration $p = 3$ and 501 function evaluations.

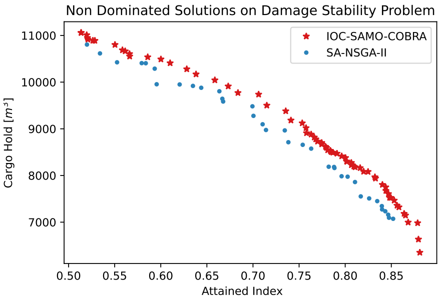
The experiments with $p = 3$ aim at investigating the potential benefit of parallelism in terms of wall-clock time provided that the corresponding number of simulator licenses is available. The HV metric is used to compare the performance of SA-NSGA-II and IOC-SAMO-COBRA. IC-SA-NSGA-II is not experimented with since one of the constraints is expensive to evaluate, and IC-SA-NSGA-II does not have a built-in option to use RBFs for that constraint.

6.3.1 Results of Cargo Vessel Design Experiment

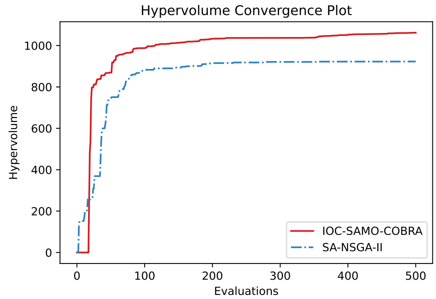
For the expensive objective (damage stability), requiring a Delftship pro simulator run, the median evaluation time was 248 seconds. In experiment 1 (IOC-SAMO-COBRA, $p = 1$, 300 evaluations), a HV of 9115 with respect to the reference point $(0, 0)$ was obtained. In experiment 2 (IOC-SAMO-COBRA, $p = 3$, 501 evaluations), the same HV was obtained in the 129^{th} iteration (after 385 function evaluations), saving a total wall-clock time of 682 minutes compared to experiment 1.

A comparison of the Pareto fronts resulting from experiments 2 and 3 (i.e., a direct comparison between IOC-SAMO-COBRA and SA-NSGA-II) is shown in Figure 6.8a, where cargo hold capacity (↑ max) is shown on the y-axis and the attained damage stability index (↑ max) on the x-axis. The Pareto front obtained by the SA-NSGA-II algorithm is dominated by the obtained Pareto front obtained by IOC-SAMO-COBRA, and the latter algorithm also finds more extreme solutions (especially for damage stability).

Figure 6.8b illustrates the convergence of the algorithms by showing the HV (measured between the Pareto fronts obtained by the two algorithms and the approximated Nadir point) over the number of function evaluations. The difference in the two Pareto fronts (Figure 6.8a) is also clearly visible in this illustration. The different behavior in the first few evaluations can be explained by the difference in the initial sampling strategies (Latin Hypercube Sampling [138] for SA-NSGA-II vs. Halton Sampling [73] for IOC-SAMO-COBRA).



(a) Comparison of Pareto fronts obtained by IOC-SAMO-COBRA and SA-NSGA-II ($p = 3$, each) on the ship design problem. x-axis: Survivability index; y-axis: Cargo hold (both ↑ max).



(b) Comparison of the hypervolume maximization progress between IOC-SAMO-COBRA and SA-NSGA-II ($p = 3$, each) on the ship design problem. x-axis: Number of function evaluations; y-axis: Hypervolume.

Figure 6.8: Obtained Pareto Frontier and Convergence plot of single hold cargo optimization problem.

6.3.2 Analysis of Cargo Vessel Design Results

The IOC-SAMO-COBRA results were further analyzed by naval architects to understand and interpret them in the light of vessel design expertise, resulting in the following observations:

- For every point on the Pareto front, the parameter that defines the tanktop height has converged to the minimum value. The algorithm learned that extra height in the double bottom of the vessel does not improve the damage stability index. The compartments above the tanktop benefited from this in terms of their size. Interestingly, this finding could be confirmed since it is also prescribed in the International Convention for the Safety of Life at Sea (SOLAS chapter II-1 part B-2 regulation 9) [81].

- The algorithm also found that a large space between the hull and cargo hold is beneficial for the damage stability criterion. This result can be explained well by the fact that a small distance between the hull and cargo hold makes it less likely for the design to survive in case of damage (flooding of the cargo hold will always lead to the loss of a single cargo hold vessel).

6.3.3 Conclusion Cargo Vessel

The single-hold cargo vessel has been optimized with the IOC-SAMO-COBRA algorithm. By using the IOC-SAMO-COBRA algorithm the available resources (3 available Delftship licences, 3 desktops) are perfectly exploited. The constraints and objective evaluations that where computationally very inexpensive have been used directly in the algorithm instead of also training an RBF. With this new functionality, it is illustrated that a significant amount of wall clock time can be saved and much better Pareto frontier approximations could be made.

6.4 Roll-on/Roll-off Ferry Hull Optimization

After an operational data analysis study and a feasibility study for alternative fuels, an initial design is made by C-Job naval Architects for a fully battery-powered Roll-on/-Roll-off (Ro-Ro) ferry with a capacity of 800 passengers [31, 32]. This vessel is designed to sail between the Saronic islands and the port of Piraeus. A render of this C-Job design is presented in Figure 6.9.

Figure 6.9: Render of Saronic Roll-on/Roll-off ferry designed and created by C-Job Naval Architects.

As this design is intended to sail powered on batteries the vessel requires an optimized hull. To accomplish the energy-efficient hull, the hull form is optimized for minimal resistance at design speed. However, three imposed restrictions limit the search space of the to-be-designed hull:

1. The hull above the waterline is only allowed to be modified marginally.
2. The displacement of the new hull design should be equal to or larger than the original hull displacement to be able to carry the passengers, vehicles, parcels, and all equipment.
3. The Longitudinal Centre of Buoyancy (LCB) should remain within 1% of the original hull to keep a good trim, heel, and intact stability without the need to move heavy components around.

With these conditions, two completely independent experiments are set up. In the first experiment experienced naval architects with hydrodynamic experience manually optimized the hull for minimal resistance. In the second experiment, the vessel is parameterized and optimized with an optimization algorithm. The hull is parameterized below the waterline varying the following seven parameters ($d = 7$):

1. transom height,
2. the transom angle,
3. the start of the midship in the longitudinal direction,
4. the shoulder location in the longitudinal direction,
5. the bulb size, (can be 0 which results in a design without a bulb)
6. the bulb width,
7. and finally the foreship width.

The design variants are parameterized and generated in the Rhino software. After generation the three computationally relatively inexpensive constraints are also calculated in the Rhino software. Finally, the hull is exported to evaluate the computationally expensive objective in the Star-CCM+ software on a High Performance Computer with 120 cores in the Microsoft Azure cloud. The constraints are computationally relatively inexpensive but still require communication between a software package they can not be called directly in the optimization algorithm as seen in Section 5.3. However, the constraints are computationally much cheaper compared to the objective. Therefore, a variant of the IOC-SAMO-COBRA algorithm is developed that models the objectives and constraints with radial basis function surrogates. Since we are optimizing a single objective problem, the hypervolume infill criteria of SAMO-COBRA is exchanged for the feasible predicted value infill criteria of the SACOBRA algorithm [13]. The most promising solution according to the infill criteria is evaluated first on the constraints, and only if the design variant is feasible according to the constraints the objective is evaluated. If the constraints are infeasible only the constraint surrogates are updated and the objective surrogate remains the same as in the previous iteration. This process continues until the algorithm has converged and no significant improvements are found for several consecutive iterations. More details and test results for this single objective optimization algorithm framework can be found in Appendix B.1 and in [147].

6.4.1 Results of Ferry Hull Optimization Experiment

After 10 initial samples, the algorithm was allowed to do 110 more evaluations. In the design of experiments, not a single feasible solution is found, however, the first solution proposed by the algorithm was feasible right away. After enough feasible solutions ($d + 1 = 8$) are found to fit a first surrogate model for the objectives, the algorithm started also optimizing the objective score. In total out of the 120 evaluations 21 feasible hulls are found. The best of the 21 feasible hulls had a 26% smaller resistance value compared to the original design.

The convergence plot of the feasible solutions are plotted in Figure 6.10. The convergence plot also shows exactly what was expected. The first 8 feasible designs still show a relatively high objective value as the algorithm is not minimizing the objective score yet but putting its attention to finding feasible solutions. After the algorithm has seen enough solutions to learn from (8 in this case since there are 7 parameters), the algorithm immediately started minimizing the objective value. The overall best hull found by the algorithm is given in Figure 6.11.

6.4.2 Analysis of Ferry Hull Results

As described earlier, besides the experiment with the algorithm, naval architects with hydrodynamic experience also optimized the hull of the ferry. In their experiment, they used their creativity to design the underwater part of the hull in a completely different way. The engineers chose a knuckle line and fitted the bulb under the bow flare instead of in front of the ship. After their choice, the naval architects used a total of 8 CFD evaluations and found a hull with almost identical objective value (a 26% reduction). The final optimized hull of the naval architects is displayed in Figure 6.12.

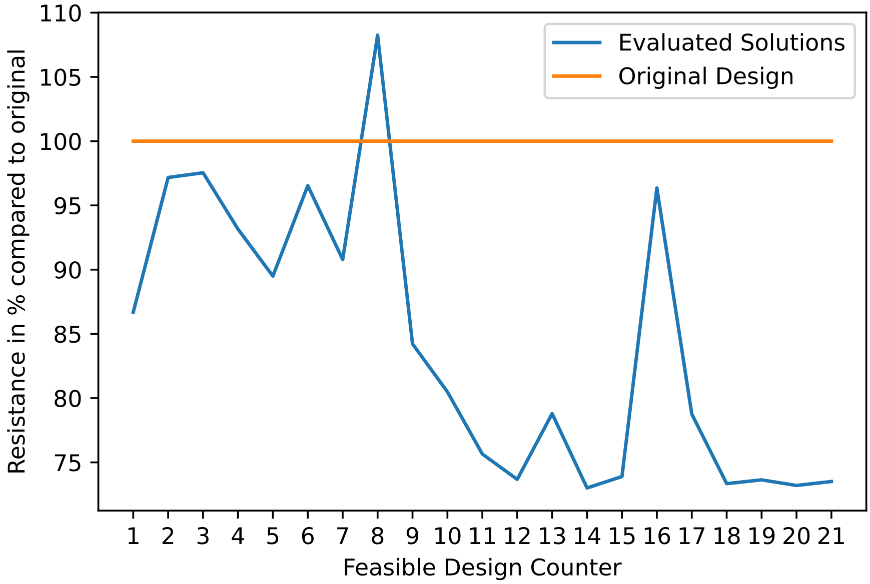


Figure 6.10: Convergence plot of the feasible solutions. Objective score in % compared to the original design.

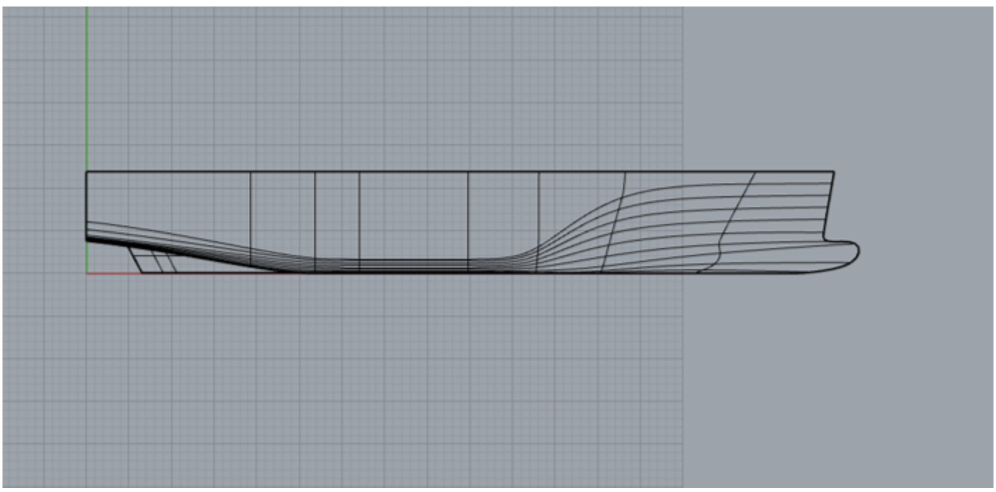


Figure 6.11: Optimized hull found by the optimization algorithm.

When compared to the design proposed by the optimization algorithm, the hull designed by the naval architects is better for a very practical reason. As this is a

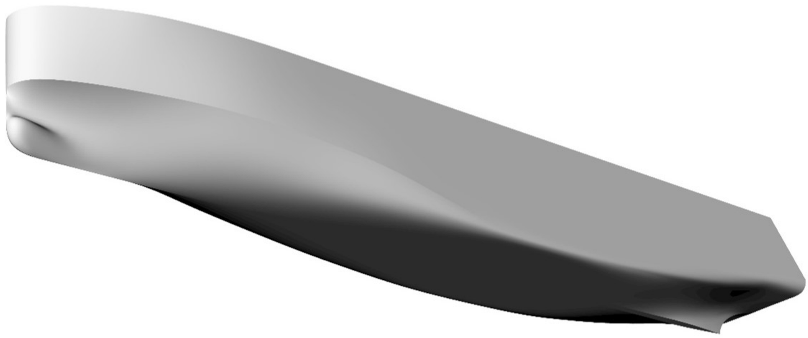


Figure 6.12: Proposed optimized hull design by naval architects.

Ro-Ro ferry, vehicles have to roll on and roll off the ship. When a bulb is in front of the ship, the ramp that the vehicles use to enter and exit the ship should be longer compared to when the bulb is under the bow flare. The downside of the design with the knuckle line is that it might be more complex to build.

6.4.3 Conclusion on Ferry Hull Optimization

The underwater part of a hull from a Ro-Ro ferry is optimized with an optimization algorithm by optimization experts and independently by naval architects with hydrodynamic experience. The best hulls from both the naval architects as the hull from the optimization algorithm showed a 26% smaller resistance compared to the original hull. The hull from the naval architects however had practical benefits compared to the hull from the naval architect. Therefore the hull of the naval architect is preferred above the hull proposed by the optimization algorithm. The conclusion that can be drawn from this is that the parameterization part of any optimization problem is the most important part when optimizing. If not all constraints are captured, the practical application is ignored, or the objective function does not perfectly represent the true goal then optimization algorithms can converge to less ideal solutions. Therefore, human experience remains important and the algorithm can only find as good solutions as the parameterization allows.

6.5 Bulb Optimization Problem

The optimization problem in this section is a refit of an existing container vessel with a container capacity around 10 000 standardized containers. Due to increasing fuel prices and more strict emission regulations, this vessel must reduce its operational speed since sailing at the design speed emits too much greenhouse gas and is way too costly. To sail efficiently at the different operational conditions, the vessel required a bulbous bow (in short bulb) refit. This bulb refit is done by cutting the current bulb off and welding a new optimally shaped bulb back in place. However, optimizing the bulb for all different loading conditions and speeds would lead to four different optimal bulb shapes. Therefore, the goal of this study is to find the bulb that performs well on (weighted)average on all four conditions.

Since the SAMO-COBRA algorithm and other algorithms discussed in this work are designed for multi-objective optimization problems and the problem at hand is a single objective problem, a single objective variant of the SAMO-COBRA algorithm has been introduced to optimize the bulb of a container vessel. The new algorithm variant uses an as small as possible initial Halton sample during the design of experiments, in every consecutive iteration the best RBF transformation strategy and kernel are chosen, and a purely exploiting infill criteria is used to find promising solutions in the adaptive sampling steps. If no improvements have been found for three consecutive iterations, the algorithm switches to an exploring infill criterion. The single objective SAMO-COBRA variant in this way is similar to the traditional Efficient Global Optimization (EGO) algorithm [85] but now with radial basis functions as surrogates and self-adjusting parameters. More details and test results for this single objective optimization algorithm framework can be found in Appendix B.1 and in [147].

The main cutting line to cut the current bulb off is at the design draft and 24 meters from the most forward part of the bulb. This allows us to change 24 meters of the fore hull and only the part below the design waterline. The bulb of the vessel is parameterized in Rhino, by varying 6 parameters that define the bulb length, height, width at two locations, and by changing the overall contour lines of the bulb. The objective is defined by a weighted sum of the required power that is needed to reach four different speeds with different loading conditions and therefore different drafts. Instead of searching for an optimal bulb for each condition, the weighted sum of the four different conditions is chosen as the objective function so that this bulb will perform well in all four conditions. However, this did mean that the complete hull had to be evaluated with RANSE calculations in the Star-CCM+ software for all four conditions. The mesh for the vessel consists of between 1.7 and 3.7 million cells depending on the speed and draft of the calculation. For this optimization challenge the new algorithm is coupled with a high-performance computer on the Microsoft Azure cloud with 120 CPUs. One RANSE calculation in Star-CCM+ of the complete hull required approximately 20 minutes, since there are 4 different conditions, one iteration required 1 hour and 20 minutes of computation time on the high-performance machine.

6.5.1 Results of Bulb Design Experiment

After a design of experiments of 7 initial Halton Samples, and 38 adaptive sampling steps, the algorithm converged to several similar optimal solutions that were all found on the boundary of the design space. However, because the solutions were found on the boundary of the design space, a second parameterization setup was made that allowed a smaller bulb in terms of height and width. The second optimization run resulted in a bulb that in total for all conditions considered had 4.8% less required power compared to the original design.

In Figure 6.13 the free surface plot is made of the original bulb (left) and the new bulb (right) in one of the four operating conditions. The color indicates the height of the water at that location (Red shows a crest, while blue shows a trough). As can be concluded from the figure, the waves that are generated with the new proposed bulb are especially in the front less extreme compared to the original hull.

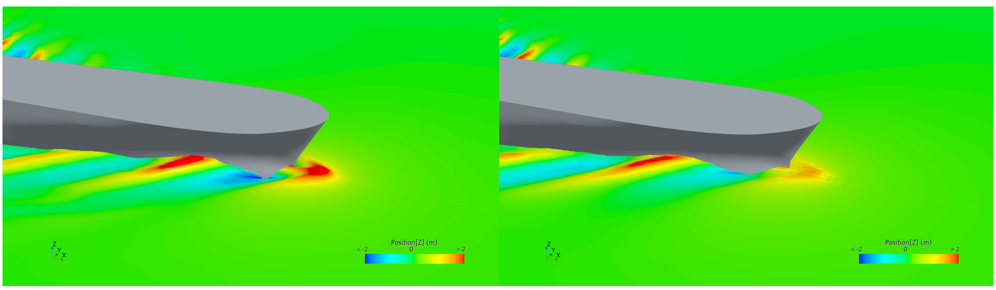


Figure 6.13: Free surface plot of the original bulb (left) and the new bulb (right). The color indicates the height of the water at that location (Red shows a crest, while blue shows a trough).

6.5.2 Analysis of Bulb Design Results

When designing completely new hulls for new build designs the hull surface is modeled with the Rapid Hull Modelling Methodology of Rhino [64]. In the rapid hull modeling methodology, the hull surface of the vessel consists of a loft that is fitted around multiple control lines. However, when refitting a bulb, wrapping a loft around a set of control lines is a bit less straightforward since the loft should exactly fit the current hull and there is only decision freedom after the cutoff line. Therefore some manual fairing was needed to create smooth and continuous hull lines without bumps and irregularities at the transition point between the hull and the bulb. After some manual fairing and checking a few extra operating conditions, the final optimized bulb showed to be able to reduce the power required the most in the slow steaming conditions and when sailing at a limited draught. This could have been expected upfront because the original bulb was designed with the principle of interacting waves for a much higher operating speed and in fully loaded conditions. This principle of interacting waves involves shaping the bulb to minimize wave resistance by strategically managing the interaction between waves generated by the bulb and the hull of the ship. The waves generated by the bulb are intended to cancel the other waves out so that in total, there is less wave resistance. Because of this principle of interacting waves, the original bulb worked best in that one condition (one draft and one speed) it was designed for.

However, When a bulb is optimized for multiple different drafts and different speeds, there is not one bulb design that generates the perfect wave for all different conditions. Therefore, the bulb that was proposed in this study does not cancel out all the waves in all conditions but it is designed to work better on the combined weighted conditions.

6.5.3 Conclusion on Bulb Optimization

To accommodate slower more energy-efficient trips a bulb refit is proposed for a container vessel with a capacity of approximately 10 000 containers. Up front, it was expected that during the optimization process a bulb would be found that generates the perfect wave to accommodate the principle of interacting waves. However, since the operating conditions to be optimized for were so different, it was difficult to find a bulb that cancels out the waves of the hull in all conditions. The newly developed single objective optimization algorithm proposed smaller than expected bulbs. The estimated performance of the small bulbs was better compared to the more traditional larger bulbs according to RANSE calculations executed with STAR-CCM+. The initial results showed that the search space was too narrow and it should be expanded to allow the algorithm to search for a bulb that was even smaller than initially thought. After the second optimization run, the optimal bulb required some manual fairing since the parameterization didn't align the new bulb perfectly with the already existing hull. Overall the optimization was a success since the hull with the new bulb showed to be 4.8% more efficient in terms of power required compared to the original hull. The largest reduction was found in the operating conditions that deviated the most from the original design condition which had a significantly higher speed.

6.6 Real World Optimization Conclusions and Future Work

Provided with the parameterization, constraint functions, and objective functions, the optimization algorithm variants were able to find feasible and optimal solutions. In the most complex design problems, feasible and optimal solutions are for naval architects often difficult to find as naval architects can't oversee all the interactions between the parameters, constraints, and objectives. The solutions found by the algorithms are however only as good as the parameterization allows, and typically after optimization require some additional practical modifications. In a few cases, it was realized that the parameterization did not lead to the expected result which therefore required a different parameterization setup, a different problem setup, or a change in objective function after which the optimization process is started again. This shows that optimization experts and naval architects should continue to work together to set up optimization problems.

In the future, more research is required in setting up the parameterization of optimization problems as usually the objective and constraint functions are quite clear but the ideal parameter and the parameter ranges that define the outcome are difficult to determine upfront. Setting up the optimization problems can be a labor-intensive and error-sensitive task. So instead of having to parameterize, define objective functions, define constraint functions, and finally optimize, it would be nice to use a generative model that could replace these steps. A start with such an approach is made with ShipHullGAN [86]. However, this thus far can only generate hulls and not complete designs including room arrangements for example.

Another future research direction that could be interesting to investigate is verifying if the proposed algorithms are also effective in other application domains.