



Propeller optimization by interactive genetic algorithms and machine learning

Downloaded from: <https://research.chalmers.se>, 2022-01-01 18:14 UTC

Citation for the original published paper (version of record):

Gypa, I., Jansson, M., Wolff, K. et al (2021)

Propeller optimization by interactive genetic algorithms and machine learning

Ship Technology Research, In press

<http://dx.doi.org/10.1080/09377255.2021.1973264>

N.B. When citing this work, cite the original published paper.



Propeller optimization by interactive genetic algorithms and machine learning

Ioli Gypa, Marcus Jansson, Krister Wolff & Rickard Benschow

To cite this article: Ioli Gypa, Marcus Jansson, Krister Wolff & Rickard Benschow (2021): Propeller optimization by interactive genetic algorithms and machine learning, Ship Technology Research, DOI: [10.1080/09377255.2021.1973264](https://doi.org/10.1080/09377255.2021.1973264)

To link to this article: <https://doi.org/10.1080/09377255.2021.1973264>



© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 05 Sep 2021.



Submit your article to this journal [↗](#)



Article views: 81



View related articles [↗](#)



View Crossmark data [↗](#)

Propeller optimization by interactive genetic algorithms and machine learning

Ioli Gypa^a, Marcus Jansson^b, Krister Wolff^a and Rickard Bensow^a

^aDepartment of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, Sweden; ^bKongsberg Maritime Sweden AB, Kristinehamn, Sweden

ABSTRACT

Marine propeller design can be carried out with the aid of automated optimization, but experience shows that such an approach has still been inferior to manual design in industrial scenarios. In this study, the automated propeller design optimization is evolved by integrating human-computer interaction as an intermediate step. An interactive optimization methodology, based on interactive genetic algorithms (IGAs), has been developed, where the blade designers systematically guide a genetic algorithm towards the objectives. The designers visualize and assess the shape of the blade cavitation and this evaluation is integrated in the optimization method. The IGA is further integrated with a support-vector machine model, in order to avoid user fatigue, IGA's main disadvantage. The results of the present study show that the IGA optimization searches solutions in a more targeted manner and eventually finds more non-dominated feasible designs that also show a good cavitation behaviour in agreement with designer preference.

ARTICLE HISTORY

Received 6 April 2021
Accepted 22 August 2021

KEYWORDS

Marine propeller design; optimization; NSGA-II; interactive genetic algorithms; machine learning; support-vector machines; cavitation constraints

1. Introduction

The selection and design of the propeller is an iterative process and involves matching towards the hull and the machinery system, considering often contradicting techno-economic requirements and objectives. The type of vessel in combination with the operational conditions play a key role in decision-making and lead to a unique propeller to be developed for each vessel. To efficiently handle this, it is important to have a well developed design process.

The selection of a good baseline propeller is normally straight-forward and performed early in the ship design process; this yields a propeller that fulfils the geometrical requirements and delivers the needed thrust. The final blade design is, however, developed late and the process runs under strict time constraints. So while the considerations in this phase involve complex phenomena related to propeller induced vibration, erosion, and noise and pressure pulses, the analysis tools must be very fast; high-fidelity simulations or experiments are not feasible. The success of a good design is thus dependent on approximate analysis tools and the expertise of an experienced designer.

In recent years, there has been a considerable effort in developing automated optimization procedures to support the blade designer. Since it concerns a multi-objective constrained design problem, the typical approach involves some stochastic population-based optimization algorithms combined with a mollified constraint handling using semi-empirical analysis of

cavitation nuisance. Although promising results have been presented it has failed to be useful in an industrial framework. The main reasons for this are that:

- The optimization algorithm is difficult to set-up to reach a converged solution; this involves the definition of the design space and the parameters that control the optimization process.
- The constraints handling fails, both in the sense that semi-empirical evaluations do not represent the reality well enough and that the number of feasible designs developed during the optimization is too low to be useful.
- The different requirements on the design are difficult to formulate into a single well-posed optimization problem; e.g. several operating conditions may need to be considered, with different cavitation nuisance requirements in each.

All things considered, it has been considered more reliable and efficient to use a traditional manual design process than struggling with managing a fully automated optimization (Vesting 2015; Kongsberg Hydrodynamic Design Team 2020).

In this paper, we propose an interactive optimization procedure as a path to overcome the issues noted above; instead of using solely quantitative constraints, the user is being called during different stages of the optimization and evaluates the solutions at that stage. This then constitutes a procedure where the designer guides a more loosely formulated stochastic

CONTACT Ioli Gypa  ioli.gypa@chalmers.se

optimization towards the desired design space. This optimization methodology is called interactive genetic algorithms and belongs to the area of interactive evolutionary computation (IEC) (Takagi 2001). A first step towards this was presented in Gypa et al. (2020), where the blade designer was called during intermediate steps of the optimization to visualize and assess specific areas of interest of the Pareto plot. The results showed it is indeed possible to steer the optimization to a desirable direction and to obtain a more refined Pareto frontier with a large number of feasible designs. Here, we extend the method by letting the blade designer visualize and assess design characteristics thereby evaluating constraints related to cavitation. More specifically, this concerns the shape of sheet cavitation on the blade that could lead to potential erosion damages, a critical constraint but typically very difficult to automate. In addition, this manual constraints evaluation is used in the early phase of the algorithm to train a machine learning algorithm.

Thus, the purpose of this study is to investigate the behaviour of a user-guided interactive optimization method as one component in an improved industrial propeller design process.

The procedure is evaluated on a simplified design scenario involving a propeller design where efficiency should be maximized and the maximum cavity volume should be minimized at the maximum continuous rating (MCR) condition. The study shows that the interactive approach leads to a well-populated Pareto frontier and generates a larger number of feasible designs compared with a fully automated approach.

This paper is organized as follows: First, a review of published efforts in automated propeller design is performed in Section 2. Section 3 then reviews different existing techniques using interactive computations. In Section 4, we present our IGA methodology coupled with a support-vector machine (SVM) model for constraint evaluation. In Section 5, the user scenario and the results are provided together with a discussion. Finally, in Section 6, the conclusions of this study are presented.

2. Automated optimization in propeller design

Due to the numerous design parameters, objectives, and constraints to consider in propeller design, the development in automated optimization processes has focused on stochastic nature-inspired population-based algorithms, as they quickly offer a set of optimal solutions spread in a wide design space (Wu et al. 2019). In blade design optimization, as a consequence of the limited time, an additional consideration is the use of analysis tools with limited

reliability and some semi-empirical evaluation of cavitation nuisance constraints.

A significant amount of research has been published by several hydrodynamic research groups on automated blade design optimization with various implementations of evolutionary algorithms combined with the utilization of faster computational tools of lower fidelity. In Berger et al. (2014), a two-stage optimization methodology for full-scale propellers working behind a ship is presented. In a first stage, an evolutionary algorithm coupled with a panel code was utilized for a multi-objective optimization problem and in a second stage, some of the best propeller designs of the first stage were chosen in order to be investigated further by a hybrid Reynolds Average Navier-Stokes (RANS) and Boundary Element Method (BEM) approach for the hull and propeller flow, respectively. The choice of the optimal designs was executed manually as an intermediate step between the two stages. Although better designs were obtained at the end of the optimization, some of these optimal solutions led to infeasible propeller designs.

Attempts at developing fully automated blade design optimization processes were presented in Vesting et al. (2016) and Vesting et al. (2016). The non-dominated sorting genetic algorithm II (NSGA-II) and particle swarm optimization (PSO) were evaluated, including extensions with meta-models, as well as different geometrical modifications and constraint handling methods. The proposed processes were evaluated for several commercial design cases, and while the outcome was satisfactory, it was still deemed inferior to the manual design. It was concluded that the manual design requires a large number of work-hours, making it difficult to obtain more than a few different designs. These designs were, however, of higher quality than the ones generated by the automated process that suffered from a large number of infeasible designs.

In Huisman and Foeth (2017), a parametric geometry model was used together with the NSGA-II for solving two propeller optimization problems with different objectives and constraints in each case, while a BEM was utilized for the calculation of the hydrodynamic performance in behind condition. This method delivered optimized designs with a high-computational cost. Note that the authors mention the importance of having a robust engineering environment for the automated optimization and that the manual design is still part of the process.

The previously mentioned issues have led to a need of having an intermediate step that would involve the blade designers in the optimization process, as suggested also in Vesting (2015). The pre-set constraints of the automated optimization guide the optimization to a desirable direction, but often they restrict the problem too much and the outcome is a

narrow design space and a small set of solutions that in fact only satisfy the constraints with slight performance improvement. In cases when there are no constraints, the solutions are found in a wide design space. However, the common problem then is that those solutions may have high performance, but most of the designs are unrealistic or infeasible, due to erroneous predictions by the fast and approximate numerical codes utilized; they would never be used in reality.

An example of such a practical optimization tool for the hydro-acoustic optimization of naval propellers was presented in Valdenazzi et al. (2019), where the optimization was done in three different levels and GAs were utilized. At first, a large design space was explored with the performance in open water as the main objective. The obtained designs from the first step defined the design space of a second round, where more objectives were set and the performance was assessed in behind conditions. Finally, a specific number of designs was chosen by the users and evaluated using a CFD solver of higher fidelity. Main priority of this optimization tool has been the low computational time and the good usability of the tool and this becomes more achievable when the users are included in the optimization process.

3. Background in interactive evolutionary computation

IEC is an optimization methodology that combines evolutionary computation with the use of subjective human evaluation. It is increasingly used for optimization problems where qualitative criteria are entailed. Traditionally, in optimization problems that use evolutionary computation with quantitative objectives/constraints, explicit functions for each one of the objectives/constraints are defined and the performance of the individuals is calculated through those functions. On the contrary, when qualitative criteria are involved, it is impossible to evaluate them through fitness functions and it becomes necessary to include the designers/engineers (they will be mentioned as users from now on) in the optimization process. This is achieved by frequently presenting some design features of the optimization individuals to the users, who subsequently evaluate those features based on their own judgement and experience. Therefore, instead of automatically calculating the performance through the fitness functions, the users assign fitness values according to the design characteristics they are presented with. Then the information elicited from the users is integrated in the optimization algorithm and utilized for the next generations of individuals.

The first IEC algorithm was proposed by Dawkins (1986) and since then, IEC has been used in various

areas, for instance in music, design, image processing, virtual reality, database retrieval, data mining and graphic arts (Takagi 2001). Nowadays, multidisciplinary design optimization processes have become more complicated and the integration of the human thought in the optimization loop is of great importance in order to acquire feasible solutions quickly (Simpson and Martins 2011). Several algorithms of the evolutionary computation, such as GAs, PSO and ant colony optimization, have been applied coupled with multiple user interactions; with the IGAs being the most commonly used. More specifically, IGAs have been extensively utilized in applications that require a small number of individuals per generation. A product design application is presented in Lee and Chang (2010), where they developed an interactive creative system for the conceptual design of a mobile phone aiming at improving the design procedure at an early stage. In their implemented IGA methodology, a hybrid fitness assignment strategy (Sugimoto and Yoneyama 2002) was applied, where the users selected only two individuals in each generation that were closer to a target design and the fitness of the remaining individuals was dependent on the distance between the selected and the remaining individuals. A similar approach was followed in Poirson et al. (2013), where IGAs were used for better understanding of the user's preferences in two product design applications. Their optimization methodology had two modes, the manual and the automatic. In the manual mode, the users evaluated all individuals of each generation, while in the automatic mode a pre-set product was randomly chosen, and the individuals of each generation were evaluated by calculating the Euclidean distances between the individuals and the target product. The latter mode was used for tuning the IGA parameters for the assessment of the code's convergence. The optimization had only nine individuals per generation and the users could assess a maximum of 30 generations before they became fatigued. Another application was presented in Yannou et al. (2008), where IGAs were used for the design of car silhouettes. Each generation had 100 individuals and the users ranked all of them with values 0–6. After a number of generations, the individuals became infeasible, so the authors suggested adding one more intermediate step, where the users would have the possibility to directly modify the designs of the individuals.

The main drawback of the IEC is that it leads easily to user fatigue (Wahde 2008). In applications where several generations of large populations are required in order to increase the performance, the users have to do numerous manual evaluations of designs and this becomes exhausting and time-consuming after a number of evaluations. This difficulty can be overcome by combining the IGAs with surrogate models (Sun et al. 2012); models that are trained with the

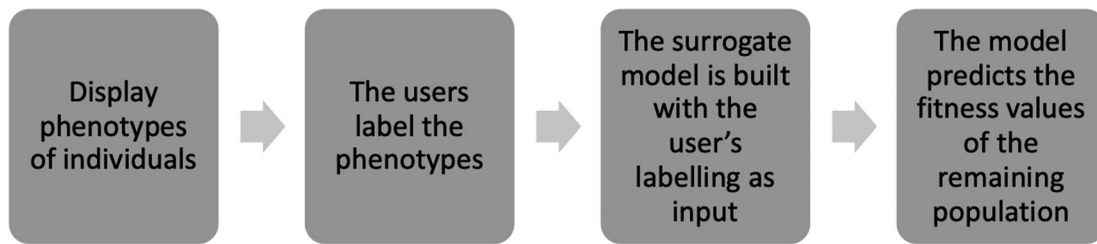


Figure 1. Structure of surrogate model.

information elicited from the users' evaluation for a part of the population and approximate the performance of the remaining non-evaluated individuals. In this manner, a system is created where it is possible to produce large populations through the IGAs and avoid the users' fatigue through the surrogate model, while the users are asked to evaluate the system's phenotypes a limited number of times. Such a system has the structure shown in Figure 1.

Various types of surrogate models can be coupled with the IGAs, such as neural networks (NNs), SVMs, and others. In Sun et al. (2012), the problem of user fatigue is eased by utilizing a model that uses two radial-basis function NNs trained by a semi-supervised learning method. Ohsaki and Takagi (1998) utilized two methods for predicting the users' evaluations, a NN and Euclidean distances, with the latter one delivering better results. Additionally, a new display method of the phenotypes was suggested for the easier user evaluations, where the individuals were presented in fitness value order. For a music application in Biles et al. (1996), where the evaluations were even more laborious, as the users had to evaluate each individual separately making the comparison impossible, surrogate models were essential. A NN with the cascade correlation technique was used, but the network overfitted the training data and the results did not converge. SVMs have been used as surrogate models in Ren and Papalambros (2011) coupled with an efficient global optimization algorithm and in Llorà et al. (2005) in combination with partial ordering concepts and non-domination from multi-objective optimization.

Blade design optimization problems require large populations of individuals in order to increase the performance and, as previously described, the involvement of the blade designers in the optimization process is necessary. Thus, in our methodology we use IGAs coupled with SVMs, used as a binary classification model, which is described in more detail in Section 4.

4. IGA methodology and SVM

A flowchart of the implemented IGA methodology is presented in Figure 2 and is described in this section. The aim of our methodology is to produce several design alternatives, present some of their design

features to the users for manual assessment and add that user information to the optimization method in order to guide it to areas of the design space that will offer more suitable designs. Before starting the optimization, a blade design is created manually, which referred to as baseline design, and the goal is to increase the performance of it in the best possible way. In order to increase the performance, several designs with small geometrical alterations are created through the optimization algorithm. The users run the optimization algorithm and at a certain point they are asked to evaluate some qualitative constraints of the problem interactively. When the users are finished with the manual evaluations, they can choose to continue the optimization process by selecting designs that they find interesting for further evolution. This procedure is iterated until desirable results are obtained.

In more detail, the optimization process begins with an optimization round using the NSGA-II algorithm (Deb et al. 2002). Every optimization process usually consists of several smaller optimization cycles, the optimization rounds, which last from initiating them with a new set-up or input until pausing them manually or finishing the loop. The users pre-define the number of generations, individuals per generation, objectives, constraints, design variables and optimization parameters. The design variables of the optimization range between a lower and an upper boundary relative to the values of the design variables of the baseline design.

After a pre-set number of generations (N), the optimization is paused and the evaluation stage follows, where there are two options. If the population of the individual designs is large, then a smaller part of the total population is presented to the users. This part consists of unique designs that are selected randomly from the total generations up to that point. If the population is small, all designs are presented. The information that is displayed on screen, in the specific scenario, is the cavity shape that has been developed on the blade of each individual propeller, as shown in Figure 3. For both small and large populations of individuals, the users evaluate the designs on the screen by rejecting or accepting them, according to their own preference and design experience. In the case of a large population, the users evaluate a smaller

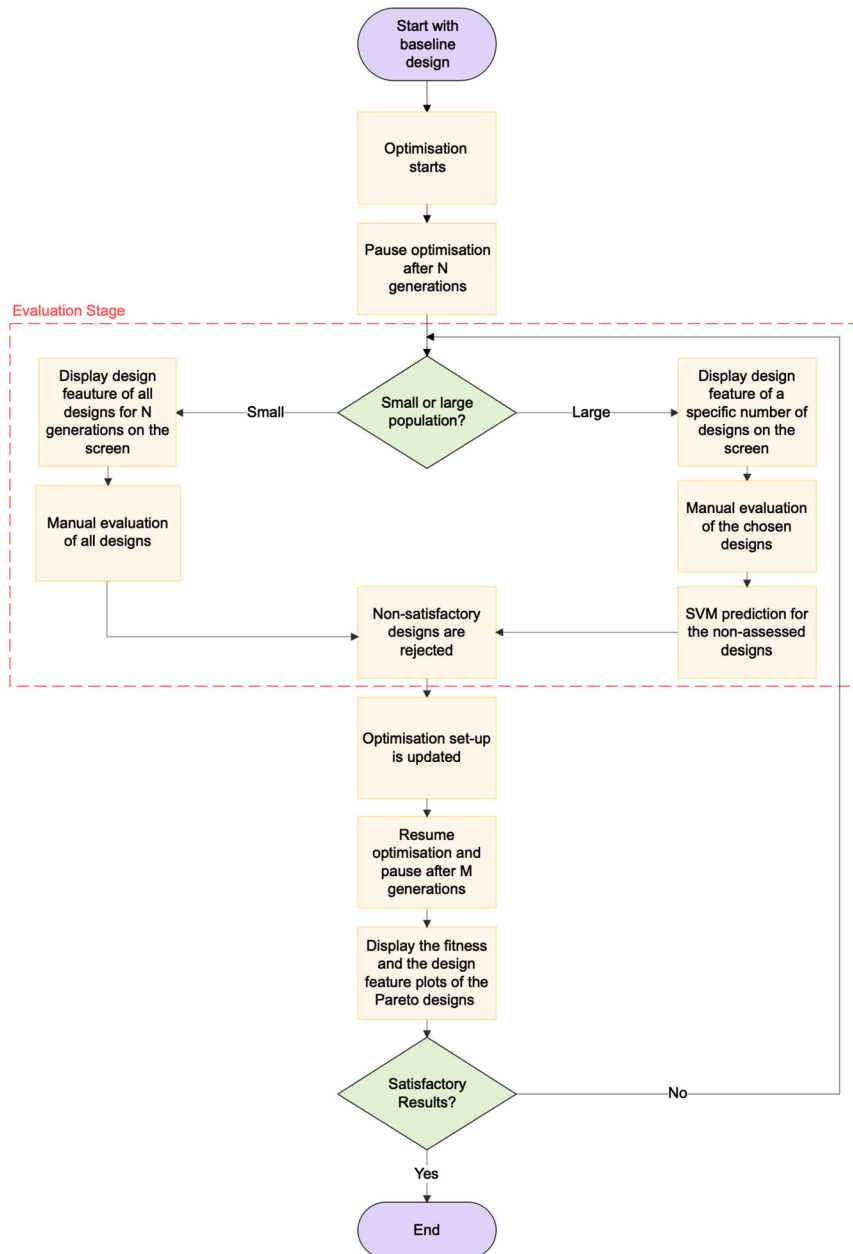


Figure 2. IGA methodology flowchart.

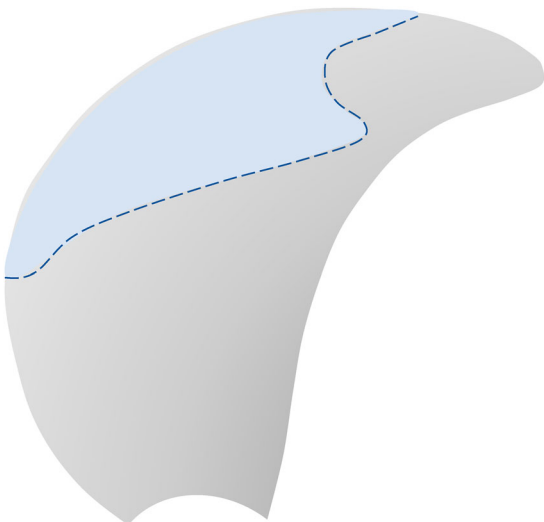


Figure 3. Cavity on blade.

part of the total population, while a prediction is made for the rest of the individuals by an SVM model. As previously mentioned, the SVM model is used for avoiding user fatigue from having to evaluate too many designs. For both cases, the non-satisfactory designs are rejected.

At the end of the evaluation stage, the users have the possibility to update the settings for the continuing optimization, e.g. number of generations, individuals per generation, objectives, constraints, design variables and optimization parameters. One of the key parameters here is the number of individuals per generation, as the first generation of the next optimization round will be comprised of only accepted designs from the previous round. If the required number of individuals for the next round is larger than the number of accepted designs, then mutation and crossover are

used for the accepted individuals to create new ones in order to fill in the missing individuals; if the number is smaller, then the accepted individuals are ranked and the non-dominated solutions are kept, similarly as in the NSGA-II; if the number is equal, then the first generation is comprised of all previously accepted designs.

When the first generation of the second optimization round has been created, the optimization is resumed. The optimization is paused again after a M number of generations and the designs of the Pareto frontier, together with the corresponding cavitation plots, are displayed on the screen. If the results are satisfying, the whole process is finished; otherwise the loop goes back to the evaluation stage. This process is repeated until desirable results have been achieved.

4.1. Support-vector machines

SVMs are broadly used in engineering as supervised machine learning methods. They were initially designed for binary classification and nowadays they are also used in applications for regression and multi-class classification (Vapnik 1995). A simple case in binary classification is shown in Figure 4. The input is data points with two features, x_1 and x_2 , that are separated in two classes. When a new data point is introduced, the SVM determine in which of the two classes it will belong. In order to attain this, the optimal hyperplane is determined that separates the classes in the best way possible and offers the maximum margin between them. A hyperplane in a n -dimensional space is a $(n-1)$ -dimensional subspace. In the figure, the hyperplane of the 2D-space is the red straight line and the margin that has to be maximized is the distance $\frac{2}{\|\beta\|}$. Also, the two parallel hyperplanes, or marginal hyperplanes, define the boundary of the margin and the 4 data

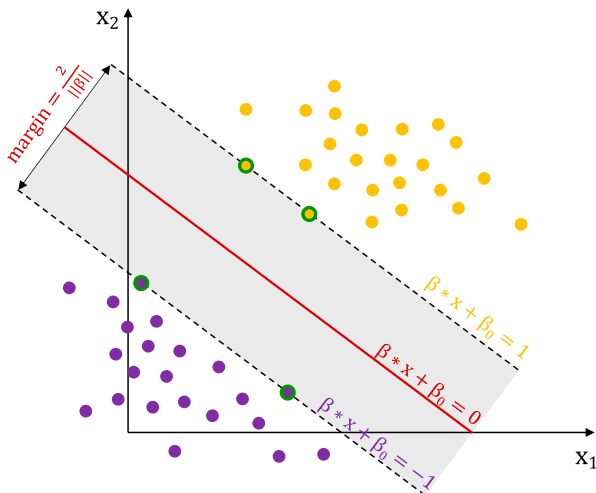


Figure 4. SVM with 2 classes that are separated from the optimal hyperplane with the largest margin.

points that are on the marginal hyperplanes are the support-vectors.

The main objective of the SVMs in binary classification is to find the optimal hyperplane that classifies the data correctly and to maximize the margin between the classes. A brief mathematical formulation for the binary classification follows according to Hastie et al. (2001). A training set (x_i, y_i) is assumed that is comprised of the multi-dimensional training data x_i , where $i = 1, \dots, n$ and $x_i \in \mathbb{R}^d$, and the class labels $y_i \in \{-1, 1\}$. The hyperplane is defined as,

$$x^T \beta + \beta_0 = 0, \quad (1)$$

where β is a unit vector with $\|\beta\| = 1$ and β_0 a real number.

Additionally, the marginal hyperplanes are defined as $x^T \beta + \beta_0 = 1$ and $x^T \beta + \beta_0 = -1$. This means that data points should be classified according to,

$$y_i(x_i^T \beta + \beta_0) \geq 1, \quad i = 1, \dots, n. \quad (2)$$

The optimal hyperplane can be found by solving the convex optimization problem with linear inequality constraints,

$$\begin{aligned} \min_{\beta, \beta_0} \quad & \|\beta\| \\ \text{subject to} \quad & y_i(x_i^T \beta + \beta_0) \geq 1, \quad i = 1, \dots, n. \end{aligned} \quad (3)$$

The above description regards data that can be entirely linearly separable, but in data sets of real-world applications linear separation is not possible. A solution to this is to use soft margins that are hyperplanes that separate the data by allowing some misclassifications with the introduction of non-negative slack variables $\xi = (\xi_1, \dots, \xi_n)$.

In this case, the optimal hyperplane can be calculated by solving the optimization problem,

$$\begin{aligned} \min_{\beta, \beta_0} \quad & \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i \\ \text{subject to} \quad & \xi_i \geq 0, \quad y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i, \quad \forall i, \end{aligned} \quad (4)$$

where C is a penalty parameter for the compromise between margin maximization and training error minimization and $\sum_{i=1}^N \xi_i$ is the training error. Finally, in the case of non-linearly separable data, kernels can be used. They map the input data into a high-dimensional space and the optimal hyperplane with the maximum margin is calculated in this space where the data can be linearly separable; this process is called Kernel trick. The Kernel function is described as,

$$K(x, x') = \langle h(x), h(x') \rangle, \quad (5)$$

where $h(x_i)$ are the transformed feature vectors. Broadly used kernels are the linear, the polynomial, the radial basis function and the sigmoid.

In this paper, we use the Scikit-learn python code (Pedregosa et al. 2011) for implementing the SVM

model in our method. As mentioned in the description of the methodology, the SVM model is utilized when the optimization algorithm is used for large population sizes, in order to reduce user fatigue. A smaller part of the total designs is chosen and their cavitation figures are displayed on screen. The users see them, observe them and evaluate them by accepting or rejecting them. Instead of labelling the designs with -1 and 1 as in the theory, in this code they are labelled with 0 and 1 . The rejected designs take the label 0 and the accepted the label 1 . This is the training process of the SVM model. The users can choose the number of times they wish to pause the algorithm and evaluate the designs. Every time new designs are evaluated, the SVM model is updated with the new information and the margin of the hyperplane is recalculated. The designs that have not been evaluated and labelled by the users, are classified in one of the two classes according to the prediction of the model. The implementation of the SVM model in the IGA methodology makes the whole optimization process faster, as it will be shown in Section 5.

5. Results and discussion

5.1. Computational tools

The hydrodynamic analysis tool that is utilized in this scenario is the MPUF-3A (He et al. 2010) code, which is based on lifting surface theory. More specifically, it follows a vortex lattice method, where the geometry of the propeller blade is represented by a lattice of discrete vortices and sources, distributed on the mean camber surface of the blade (Vesting 2015). In addition to the fundamental assumptions of potential flow theory, for the strength of the vortices, the kinematic boundary condition on the blade surface must be considered, where the wetted portion of the blade surface is impermeable to the fluid, and the Kutta condition at the trailing edge, where the flow should leave the trailing edge in a tangential direction (Lee 1979).

The cavity prediction is based on an iterative process, according to Kerwin et al. (1986), where for a 2D cavitating profile section, the cavitation is represented by sources and the source strengths must be solved for each time step. For the 3D solution, the cavity length must be adjusted for all sections at one time step. More specifically, radial stripes of the blade must be solved until convergence is accomplished, starting from the hub until the tip of the blade and back. Each blade section depicts a set of stripes in the flow field by combining the undisturbed inflow and the induced flow of the other stripes. The solutions are computed for only one blade, the key-blade, with the aim to save computational cost. Subsequently, it is assumed

that the strengths of the vortices and sources on the other blades, correspond to those that were calculated for the key-blade (Vesting 2015). In addition to this, a fixed wake is being taken into consideration. MPUF-3A includes also the effect of the hub, the wake alignment in circumferentially averaged inflow with an arbitrary shaft inclination angle and the nonlinear thickness loading coupling (Kinnas et al. 2003). An example of how the cavity is being shaped on the blade of one propeller design in 9 consecutive time steps, is presented in Figure 5, where the cavity shape for each 2D stripe is plotted. Finally, it should be emphasized that it is possible to combine the developed methodology with other computational tools instead of the MPUF-3A code.

5.2. User scenario

We have developed a user scenario for the design of a fixed pitch propeller for a single-screw car-carrier. There are six design parameters: the pitch over the propeller diameter and the maximum camber, at 14%, 70%, and 100% (pitch) or 95% (camber) blade radii and range $\pm 15\%$ from the values of the baseline design. Additionally, there are two objectives: the maximization of the efficiency and the minimization of the maximum cavity volume at the MCR condition, while operating in a fixed wake. The input for the wake has been computed from model testing. There is one quantitative constraint, a pitch adjustment based on given threshold values for propeller thrust ($\pm 2\%$ from baseline thrust).

5.2.1. User evaluations

For this problem, the risk of cavitation erosion is a constraint that is difficult to quantify numerically but we need to rely on user experience. More specifically, the user evaluates the designs, solely based on one criterion: the cavity shape on the blade of the designs displayed on screen, when they have reached the maximum cavity volume, as shown in Figure 6. After communication with blade designers (Kongsberg Hydrodynamic Design Team 2020), it has been concluded that based on design experience and knowledge, it is sufficient to present the cavity only at the time step where there is maximum cavity volume. In the scenario studied here, cavitation shapes as in propeller (a) are preferred, because the cavitation grows and collapses smoothly when the blade passes through the wake. On the contrary, propellers (b) and (c) would certainly be rejected, since there is large cavity thickness on the tip of the blade, which could potentially lead to cavitation erosion. It should be noted, that the optimization is driven by the objectives, but in order to consider the



Figure 5. Cavity shape on blade in 9 time steps.

cavitation behaviour as well, the user evaluation of the cavity is used as a qualitative constraint in the optimization process. Thus, even designs with low performance, in terms of objectives, are being displayed on screen, so that the user evaluates their cavity shapes, with the aim to reach a Pareto frontier with designs with high performance and satisfactory cavitation behaviour. In the following subsections, the results of the IGA methodology and the SVM model are presented and discussed and a comparison with a fully automated approach is done.

5.3. Interactive optimization

In order to investigate the capabilities of the IGA methodology, the user scenario was examined with four sequential optimization rounds, where the users were asked to interact with the code in between these rounds for three times. The following procedure was developed: an optimization round was run and paused after a certain number of generations. The users were asked to evaluate the cavitation of the unique designs and they would accept or reject them according to their own preference. Subsequently, the

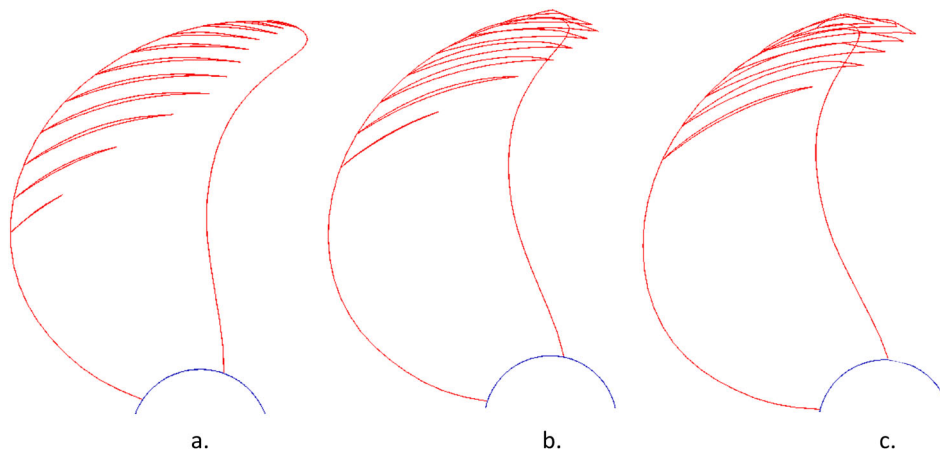


Figure 6. Cavity shape of three propellers at the angle of the maximum cavity volume.

users would choose some or all the accepted designs in order to construct the first generation of the following optimization round and at the same time they had the option to change the optimization set-up. Similarly, at the end of the next two small optimization rounds the same approach was followed, while the fourth optimization round was run for more generations.

Figure 7 shows the plots of the two objectives for all four optimization rounds and the designs are divided in accepted and rejected with green triangle markers and red x markers, respectively. In the first round, which had in total 40 designs, of which 19 unique designs, the users accepted 6 and rejected 13. In the second round, out of the 25 unique designs, 23 were accepted and 2 were rejected. The third round was a bit longer than the first two, with total 100 designs, of which 55 were unique designs, where 51 were accepted and 4 were rejected. The final round had 960 designs, out of which 213 were unique and the users accepted 185 and rejected 28 designs.

As shown in Figure 7(a), the optimization started with a small population of mostly rejected designs and ended up, as depicted from Figure 7(d), with a large population, where the majority of the individuals were accepted. It was decided to run first three small optimization rounds, so that the users were enabled to interact frequently with the optimization code and their information led to the final round where the designs fulfilled the objectives without being infeasible. It should be mentioned that in the last round all designs were in the end evaluated manually, in order to present a comprehensive overview of the optimization's progress. In most cases, when the last round finishes, the blade designers are only interested in the non-dominated solutions, thus they would manually evaluate only the interesting designs of the Pareto frontier. Figure 8 shows the evolution of the process with respect to the fitness of the objectives from all four rounds. It is observed that we manage to obtain a detailed final Pareto frontier with a large decrease in the cavity volume, yet with a slight increase in the efficiency, when compared to the values of the baseline design.

5.4. Automated optimization

We also performed an automated optimization for the same problem according to Vesting et al. (2016). The same user scenario was used, in order to be able to compare the two approaches, as it will be described in 5.5. For this type of optimization, the users pre-defined the design and optimization parameters, similarly as in each optimization round of the interactive approach, but for more generations. Then, the optimization was run and at the end the users assessed the designs of the Pareto frontier. Again, here we manually evaluated all designs to show the complete picture

of the optimization's evolution and the feasibility of the designs. A total of 1100 designs were produced, no cavitation-related constraints were set and the same thrust constraint was set as in the interactive optimization.

5.5. Comparison between interactive and automated approaches

A comparison of the two approaches is carried out in this section. The results from all four rounds of the interactive approach have been collected in one plot for the two objectives, presented in Figure 9(a) and the corresponding plot of the automated optimization is shown in Figure 9(b). The automated algorithm generates 436 unique designs, while the interactive results in 236 unique designs, due to the fact that the user evaluations constrain the problem and guide it in a direction according to their preference, while the automated optimization in this case is completely unconstrained. The obtained solutions of the automated approach are more spread, but at the same time, they are more irrelevant to the non-dominated solutions. Regarding the evaluation of the solutions, the two plots show that the interactive approach offered 189 accepted and 47 rejected unique designs, while the automated approach offered 258 accepted and 178 rejected unique designs. It is also evident that the majority of the automated optimization's diverse solutions are rejected, something that leads to unnecessary increased computational cost. As presented on the Pareto frontiers of the two approaches in Figure 10(a,b), the interactive optimization gives 49 accepted and 10 rejected designs and the automated optimization gives 38 and 7, respectively. This means that the users would obtain 11 more non-dominated solutions – useful propeller designs – at the end of the interactive optimization.

The design space of the two approaches is demonstrated in the parallel coordinates' visualization of Figure 11. The first six vertical axes represent the range of values of the six design variables of the problem, the next two axes show the values of the two objectives and in the last axis the binary user evaluation of the individuals is presented. Thus, each line represents one design and shows the values of its design variables, objectives and finally if it was accepted or rejected. The green and red colours of the lines depict the accepted and rejected designs, respectively. This type of plot can become a useful tool for the blade designers, as it is interactive and they can filter the range of the variable they are interested in.

The first observation is that the design variables of the interactive approach have not been explored as broadly as in the automated process. The reason

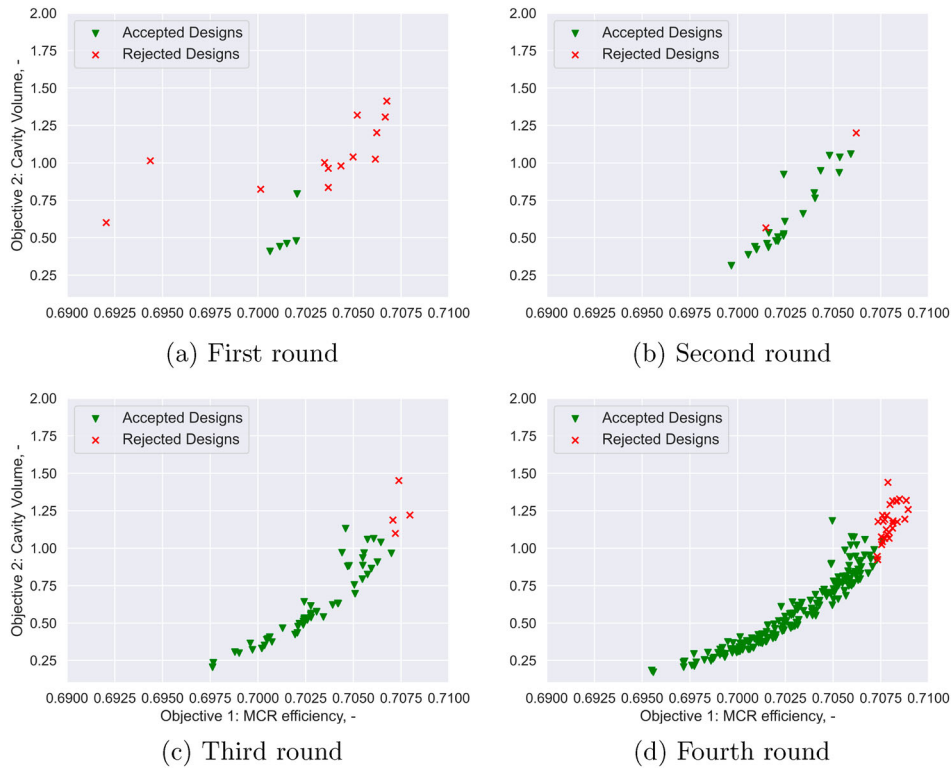


Figure 7. Accepted and rejected designs of four optimization rounds. (a) First round, (b) second round, (c) third round, (d) fourth round.

for this is twofold: the automated approach produces more unique designs, while for the interactive approach the users accept specific designs that belong to specific areas of the design space and these are promoted to the next generations. These areas of the design space are more evident in

Figure 12(a). It is observed that there is a tendency to accept designs with a pitch over propeller diameter distribution that has values over 0.9 at 70% radius and values lower than 0.75 at 100% radius, combined with camber distribution that increases at the tip of the blade (higher than 27 mm). The

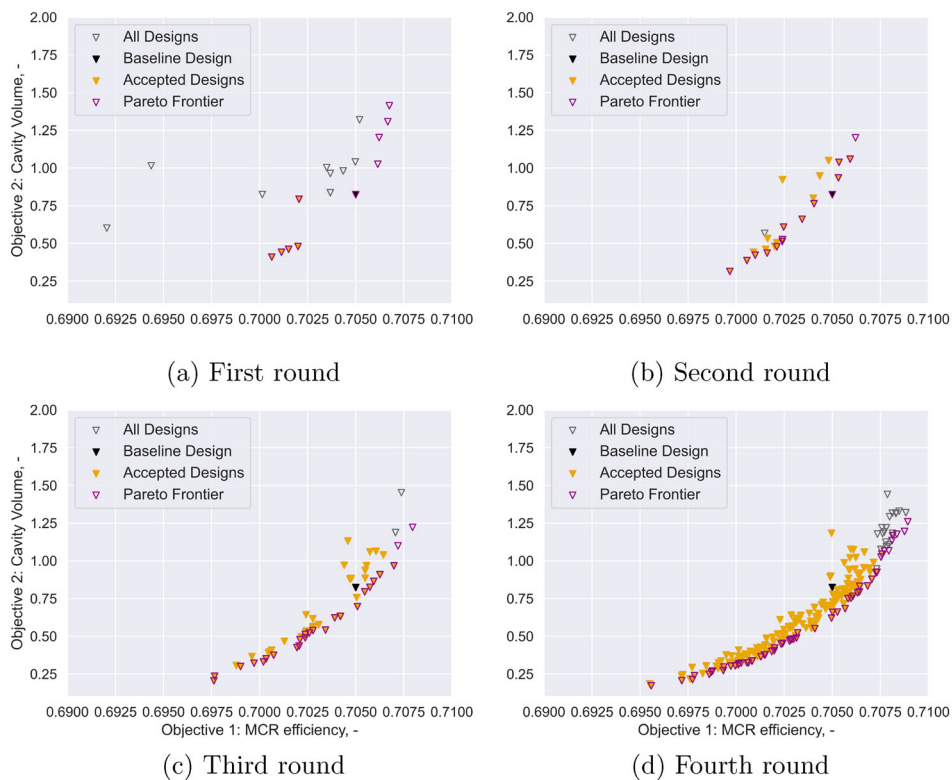


Figure 8. Evolution of four optimization rounds with IGA. (a) First round, (b) second round, (c) third round, (d) fourth round.

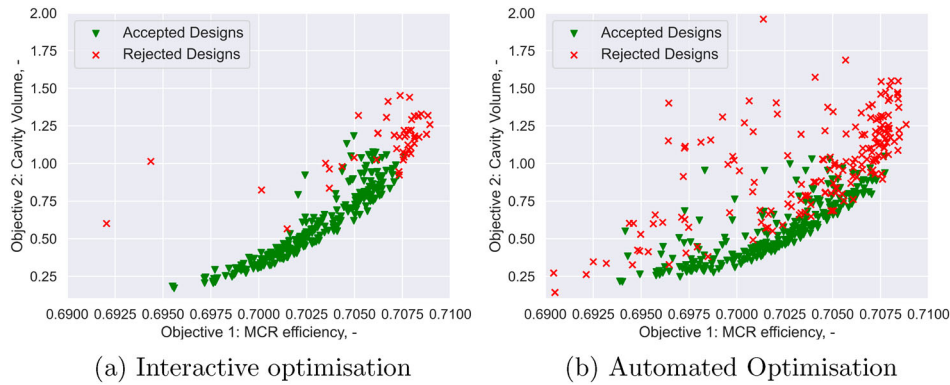


Figure 9. Accepted and rejected designs of two approaches. (a) Interactive optimization, (b) automated optimization.

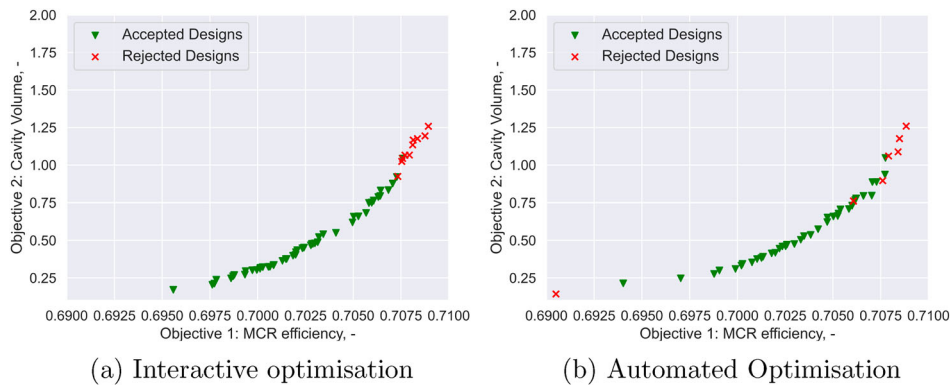


Figure 10. Final Pareto frontiers. (a) Interactive optimization, (b) automated optimization.

same trend is observed in Figure 12(b), but the algorithm has not searched in areas where the camber at 95% radius is as increased as in the interactive approach. More specifically, 26 individuals, where the camber at 95% radius has values higher than 30 mm ended up being in the final Pareto frontier of the interactive optimization. Finally, by comparing the objectives of the two approaches, their values are at the same level, but the limited exploration of the design space becomes the benefit of the interactive approach, since it gives eventually more non-dominated solutions of equal fitness to the designers.

5.6. Interactive optimization with SVM implementation

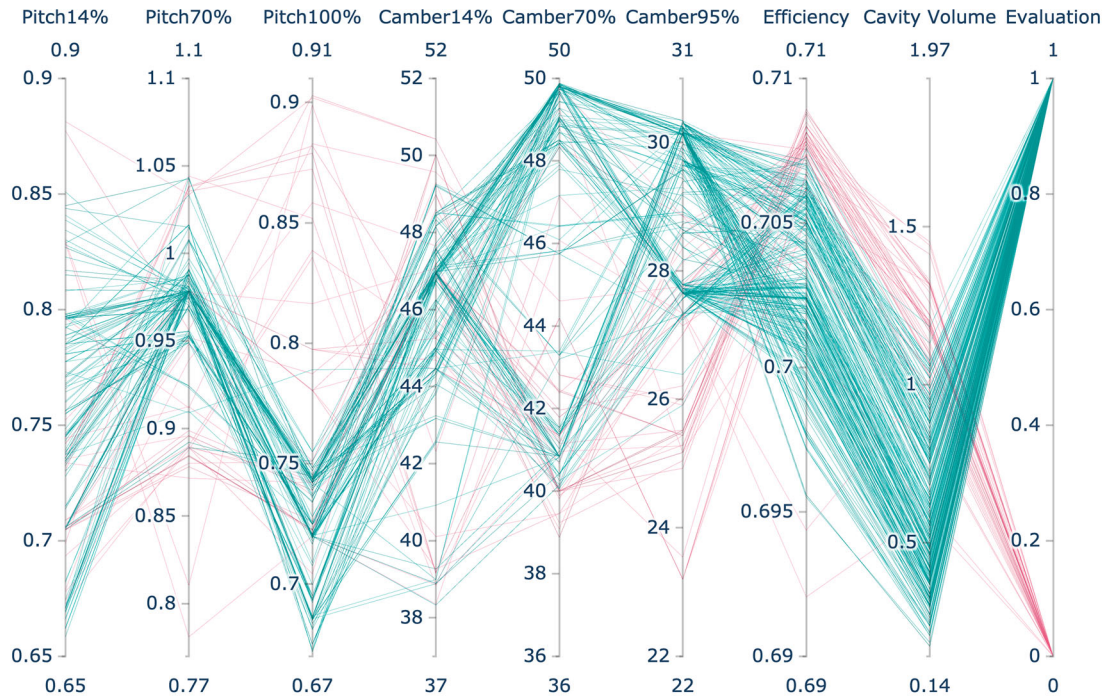
In Section 5.3, the results of the IGA methodology were presented without involving the SVM model. As it was described, after the end of the fourth optimization round, all designs were manually evaluated by the users, with the purpose of showing precisely how the code evolves according to the user preference. However, this is a labour-intensive process for the designers and can be improved by using the SVM model during the last stage of the optimization. The user manually evaluated 99 designs in the first three optimization rounds by the binary choice of accepting

or rejecting them; this corresponds to a labelling with 1 and 0 respectively in the code. Furthermore, the input to the SVM consists of this binary labelling together with the following cavitation parameters: maximum non-dimensional cavity volume at the most critical angle, chord-wise centroid harmfactor, maximum non-dimensional length of each cavitating blade section, cavity closure line harmfactor, cavitation thickness at the three outermost blade radii and non-dimensional cavity change. More information about these cavitation parameters can be found in Vesting et al. (2016). The fact that the 99 user evaluations take place in three different steps, with the optimization running in between them, causes less fatigue to the users. The input trains the SVM model and when the final round finishes, the model classifies the new designs as accepted or rejected. The total number of unique designs were 236, so the train data are the 42% of the total and the test data the remaining 58%. For the validation of the accuracy of the SVM model, the model's predicted classification was compared to the manual evaluations of 5.3. As shown in Table 1, by using either the linear or the polynomial Kernel the predictability is very high, 98.5% and 97%, respectively.

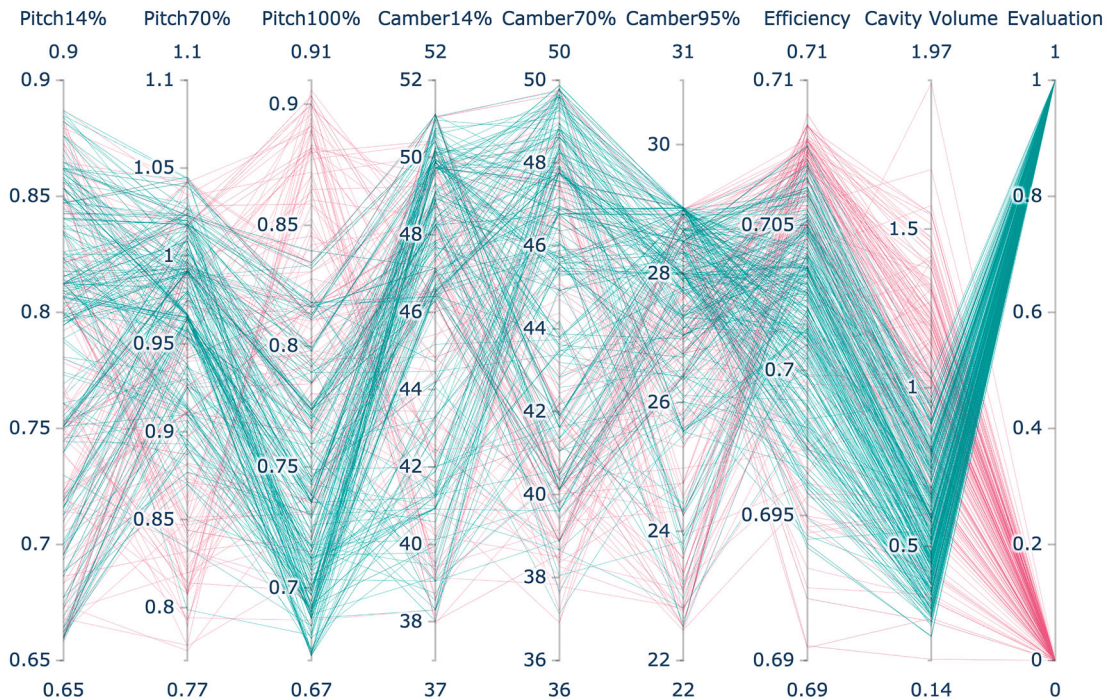
In Figures 13 and 14, the class separation is presented through decision function contours for linear and polynomial kernels, respectively. The input of

the SVM model is 8-featured, so it is impossible to present the 8-dimension class separation in a 2D plot. Thus, for the demonstration of the class separation, the principal component analysis (PCA) is used here. PCA is a multivariate technique that reduces the dimensionality of a large data set with several possibly correlated variables, by

transforming it to a new set of variables that are uncorrelated and at the same time they maximize the variance. These new variables are the principal components and their calculation reduces to the solution of an eigenvalue/eigenvector problem Jolliffe (2002). Figures 13(a) and 14(a) show the class separation according to the manual

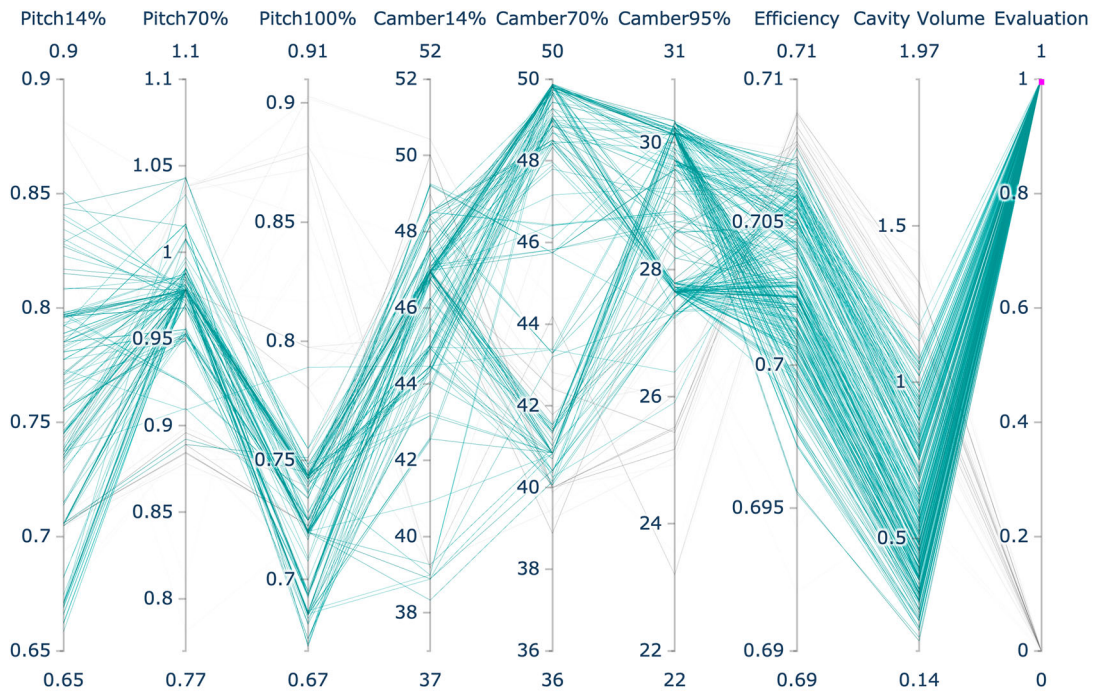


(a) Interactive optimisation

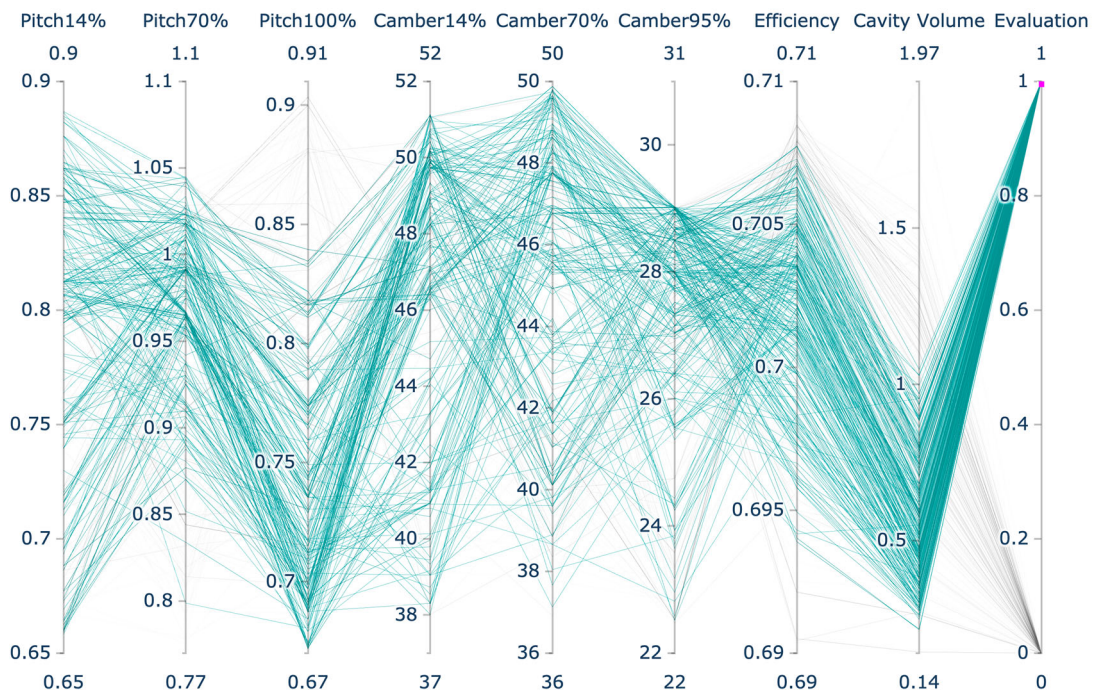


(b) Automated Optimisation

Figure 11. Parallel coordinate visualization of design variables, objectives, evaluations. (a) Interactive optimization, (b) automated optimization.



(a) Interactive optimisation



(b) Automated Optimisation

Figure 12. Parallel coordinate visualization – accepted designs. (a) Interactive optimization, (b) automated optimization.

evaluations. The penalty parameter C is set equal to 1 and there are two misclassified points (outliers) from the separating hyperplane. The higher the

Table 1. SVM predictability of last optimization round in interactive approach.

% of Training data	Predictability	
	Linear Kernel	Polynomial Kernel
42	0.985	0.97

value of C parameter, the more training points will be classified correctly, but the maximization of the hyperplane will become more complicated. In Figures 13(b) and 14(b), the predicted values are presented that have two and four outliers, respectively. It is evident that the predictability of both models is satisfactory.

Additionally, the data of the automated approach were used to do a sensitivity analysis on the SVM

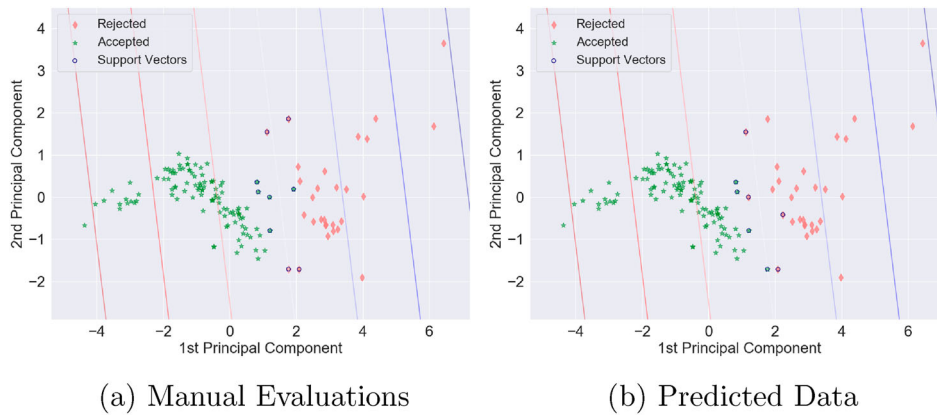


Figure 13. Class separation in SVM model with PCA – Linear Kernel. (a) Manual evaluations, (b) predicted data.

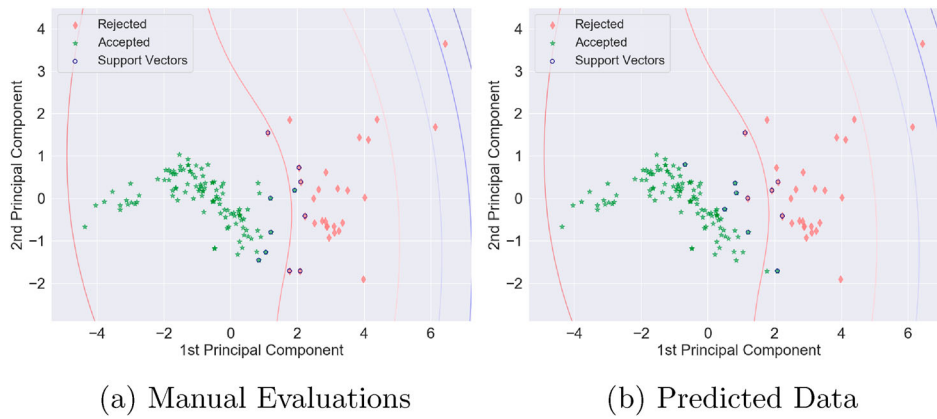


Figure 14. Class separation in SVM model with PCA – Polynomial Kernel. (a) Manual evaluations, (b) predicted data.

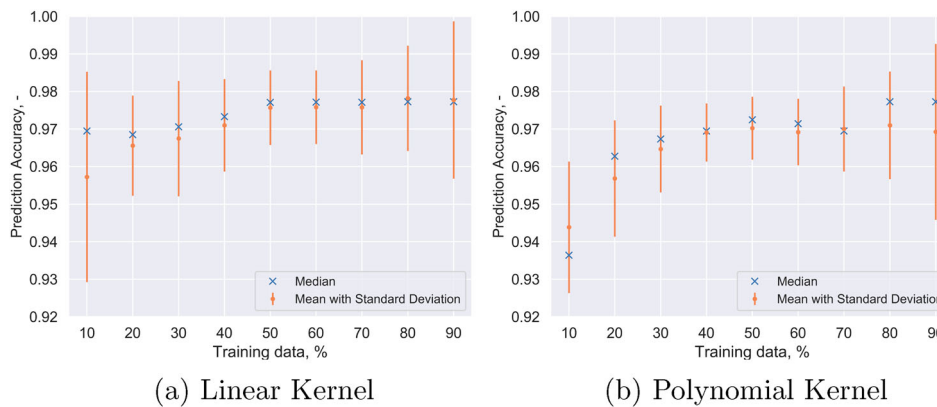


Figure 15. Summary statistics of SVM prediction accuracy. (a) Linear kernel, (b) polynomial Kernel.

model's prediction accuracy by comparing the SVM prediction with the manual evaluations. This is shown in the summary statistics of [Figure 15](#) for the linear and polynomial kernels. Different training data were chosen randomly as subsets of the whole data set and were input in the SVM; 9 different sizes of training data were considered and the process was repeated 150 times for each size. Both kernels have good accuracy, with the linear kernel being a bit higher. It is observed that when the training data is 50–90% of the data set the mean accuracy is

approximately the same, with the lower standard deviation being at 50–70% of the data set. According to these results, for the specific problem, the users should choose to manually evaluate 50% of the unique designs, in order to have as few manual evaluations as possible with accurate prediction.

6. Conclusions

In this paper, we developed a novel, general user-computer interactive optimization methodology for the

design of ship propellers. Specifically, a fixed-pitch propeller for a single-screw car-carrier was used in this study. Furthermore, IGAs were utilized and were combined with an SVM model for solving the problem of user fatigue that is the main disadvantage of the IEC. According to our methodology, the users start a NSGA-II optimization and in different stages they can pause it and visualize and assess images that show the cavity shape that has been developed on the blade at the most critical angle. The user's evaluations separate the designs in accepted and rejected and the optimization code evolves by only using the accepted solutions. Through the proposed methodology, the results show that the users obtain at the end of the optimization a detailed Pareto frontier, with non-dominated and feasible design solutions that have a good cavitation behaviour in line with the designers' preference. When compared to the results of the automated approach, both have approximately the same objective fitness, but in the automated approach, the design space is searched more broadly; at the same time, most of those diverse solutions are rejected by the users, as they do not fulfil the cavitation requirements. Thus, the interactive approach finds more solutions in a more targeted manner. Regarding the SVM model, it is needed when the populations are large in order to reduce user evaluations and its predictability accuracy proved to be very high for both linear and polynomial kernels.

The proposed methodology is a start for involving the designers in the optimization loop of marine propeller designs problems and using the designer knowledge on obtaining better designs faster. However, we believe that the benefits of our interactive approach will be more evident in a more complex scenario with more objectives and operational conditions. In such a scenario with contradictory objectives that make compromises necessary, the designer's assessments should guide the code to the correct design space areas. Additionally, a more technical point for further development is the way the designs are presented to the users for assessment. Currently, they are presented to the users one by one, but if more designs would be displayed simultaneously, then the users would be able to compare different designs at once. The presentation of designs affects also how fast the user fatigue is developed and it defines whether a population is small or large for the IGAs. For example, manual evaluations of total 100 designs that take place in groups of 10, will lead to less fatigue than executing 100 single evaluations. Lastly, it would be potentially beneficial to investigate how the data of large marine propeller databases can be used for training SVMs or other machine learning models with the purpose of assisting blade designers to design blade designs of higher quality and in less time.

Abbreviations

BEM	Boundary Element Method
IEC	Interactive Evolutionary Computation
IGA	Interactive Genetic Algorithm
MCR	Maximum Continuous Rating
NN	Neural Network
NSGA-II	Non-dominated sorting genetic algorithm II
PCA	Principal Component Analysis
PSO	Particle Swarm Optimisation
RANS	Reynolds Average Navier-Stokes
SVM	Support-Vector Machine

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Funding for this study was provided by Chalmers University of Technology Foundation for the strategic research project Hydro- and aerodynamics; by the Swedish Transportation Agency via Lighthouse through the SailProp project; and by Kongsberg Maritime Sweden AB through the University Technology Centre in Computational Hydrodynamics hosted at the Department of Mechanics and Maritime Sciences at Chalmers.

References

- Berger S, Druckenbrod M, Pergande M, Abdel-Maksoud M. 2014. A two-stage optimisation method for full-scale marine propellers working behind a ship. *Ship Technol Res.* 61(2):64–79.
- Biles J, Anderson P, Loggi L. 1996. Neural network fitness functions for a musical IGA. *Proceedings of the International ICSC Symposium on Intelligent Industrial Automation (IIA'96) and Soft Computing (SOCO'96)*; Reading, UK.
- Dawkins R. 1986. *The blind watchmaker*. New York (NY): W.W. Norton & Company.
- Deb K, Pratap A, Agarwal S, Meyarivan T. 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput.* 6(2):182–197.
- Gypa I, Bensow R, Wolff K, Gustafsson R. 2020. Interactive evolutionary computation for propeller design optimization of wind-assisted vessels. *AIAA AVIATION 2020 FORUM*, Virtual Event.
- Hastie T, Tibshirani R, Friedman J. 2001. *The elements of statistical learning*. (Springer Series in Statistics), New York: Springer Inc.
- He L, Chang S, Kinnas S. 2010. Mpuf-3a version 3.0. Technical Report, University of Texas.
- Huisman J, Foeth E. 2017. Automated multi-objective optimization of ship propellers. *Proceedings of the Fifth International Symposium on Marine Propulsors (SMP'17)*; Espoo, Finland.
- Jolliffe I. 2002. *Principal component analysis*. New York: Springer-Verlag.
- Kerwin J, Kinnas S, Wilson M. 1986. Experimental and analytical techniques for the study of unsteady propeller sheet cavitation. *Proceedings of the 16th Symposium on Naval Hydrodynamics*; Berkeley (CA).
- Kongsberg Hydrodynamic Design Team. 2020. Private Communication.
- Lee C-S. 1979. Prediction of steady and unsteady performance of marine propellers with or without cavitation by

- numerical lifting-surface theory [PhD thesis]. Massachusetts Institute of Technology.
- Lee J-H, Chang M-L. 2010. Stimulating designers' creativity based on a creative evolutionary system and collective intelligence in product design. *Int J Ind Ergon.* 40 (3):295–305.
- Llorà X, Sastry K, Goldberg D, Gupta A, Lakshmi L. 2005. Combating user fatigue in igas: partial ordering, support vector machines, and synthetic fitness. Proceedings of the 7th annual conference on Genetic and Evolutionary Computation (GECCO '05); Washington (DC).
- Ohsaki M, Takagi H. 1998. Improvement of presenting interface by predicting the evaluation order to reduce the burden of human interactive EC operators. Proceedings of 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218) (SMC'98); San Diego (CA). Vol. 2, p. 1284–1289.
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. 2011. Scikit-learn: machine learning in Python. *J Mach Learn Res.* 12:2825–2830.
- Poirson E, Petiot J-F, Boivin L, Blumenthal D. 2013. Eliciting user perceptions using assessment tests based on an interactive genetic algorithm. *J Mech Des.* 135(3):031004.
- Ren Y, Papalambros PY. 2011. A design preference elicitation query as an optimization process. *J Mech Des.* 133 (11):111004.
- Kinnas SA, Lee H, Young YL. 2003. Modeling of unsteady sheet cavitation on marine propeller blades. *Int J Rotating Mach.* 9:671–948.
- Simpson TW, Martins JR. 2011. Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop. *J Mech Des.* 133(10):153.
- Sugimoto F, Yoneyama M. 2002. Hybrid fitness assignment strategy in IGA. Proceedings of the 2002 IEEE Workshop on Multimedia Signal Processing; St.Thomas (VI). p. 284–287.
- Sun X, Gong D, Zhang W. 2012. Interactive genetic algorithms with large population and semi-supervised learning. *Appl Soft Comput.* 12:3004–3013.
- Takagi H. 2001. Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proc IEEE.* 89(9):1275–1296.
- Valdenazzi F, Conti F, Gaggero S, Grassi D, Vaccaro C, Villa D. 2019. A practical tool for the hydro-acoustic optimization of naval propellers. Proceedings of the 8th Conference on Computational Methods in Marine Engineering (Marine 2019); Gothenburg, Sweden.
- Vapnik VN. 1995. The nature of statistical learning theory. Berlin: Springer-Verlag.
- Vesting F. 2015. Marine propeller optimisation – strategy and algorithm development [PhD thesis]. Chalmers University of Technology.
- Vesting F, Bensow R, Johansson R, Gustafsson R, Costa NA. 2016. Procedure for application-oriented optimisation of marine propellers. *J Mar Sci Eng.* 4(4):83.
- Vesting F, Gustafsson R, Bensow R. 2016. Development and application of optimisation algorithms for propeller design. *Ship Technol Res.* 63:50–69.
- Wahde M. 2008. Biologically inspired optimization methods: an introduction. Boston (MA): WIT Press.
- Wu G, Mallipeddi R, Suganthan PN. 2019. Ensemble strategies for population-based optimization algorithms – a survey. *Swarm Evol Comput.* 44:695–711.
- Yannou B, Dihlmann M, Awedikian R. 2008. Evolutive design of car silhouettes. Proceedings of the 34th Design Automation Conference (DAC 2008); New York (NY).