Controllable pitch propeller optimization through meta-heuristic algorithm

Published Version:
Bacciaglia, A., Ceruti, A., Liverani, A. (2021). Controllable pitch propeller optimization through meta-heuristic algorithm. ENGINEERING WITH COMPUTERS, 37(3), 2257-2271 [10.1007/s00366-020-00938-8].

Availability:
This version is available at: https://hdl.handle.net/11585/717058 since: 2024-05-28

Published:
DOI: http://doi.org/10.1007/s00366-020-00938-8

Controllable pitch propeller optimization through meta-heuristic algorithm

Antonio Bacciaglia [1], Alessandro Ceruti [2], Alfredo Liverani [3]

School of Engineering and Architecture, DIN Department, University of Bologna, Viale Risorgimento 2, Bologna, Italy

Abstract

This paper describes a methodology to design and optimize a controllable pitch propeller suitable for small leisure ship boats. A proper range for design parameters has to be set by the user. An optimization based on the Particle Swarm Optimization algorithm is carried out to minimize a fitness function representing the engine's fuel consumption. The OpenProp code has been integrated in the procedure to compute thrust and torque. Blade's geometry and tables about pitch, thrust and consumption are the main output of the optimization process. A case study has been included to show how the procedure can be implemented in the design process. A case study shows that the procedure allows a designer to sketch a controllable pitch propeller with optimal efficiency; computational times are compatible with the design conceptual phase where several scenarios must be investigated to set the most suitable for the following detailed design. A drawback of this approach is given by the need for a quite skilled user in charge of defining the allowable ranges for design parameters, and the need for data about the engine and boat to be designed.

Keywords:
Design Propeller, Particle swarm optimization, Controllable pitch propeller, CAD

Nomenclature

Symbol	Quantity	Unit	Symbol	Quantity	Unit
A_E/A_O	Blade area ratio		n	Rotational speed	$1/s$
β_i	Initial stagger angle	deg	P/D	Pitch distribution along blade	
β_2	Stagger angle for intermediate velocity	deg	P_T	Thrust power	W
β_1	Stagger angle for lowest advance velocity	deg	P_D	Delivered power	W
c/D	Chord distribution along blade		Q	Torque	Nm
D	Diameter	m	RPM	Rotational speed	$1/min$
D_{hub}	Hub diameter	m	ρ	Water density	kg/m^3
D_opt	Optimum propeller diameter	m	σ_N	Cavitation number	
η	Propeller efficiency		t	Deduction factor	
FC	Fuel consumption	kg/h	T	Generated thrust	N
f/c	Camber distribution along blade		T_{des}	Desired thrust	N
Γ	Vortex circulation		$\%_{th}$	Throttle percentage	
J	Advance ratio		V_a	Volumetric mean inflow velocity	m/s
K_T, K_Q	Thrust and Torque propeller coefficients		V_s	Advance ship speed	m/s
k_{cav}	Cavitation penalization coeff.		Z	Blade number	
k_{stress}	Maximum stress penalization coeff.				
k_{thrust}	Generated thrust penalization coeff.				

1. Introduction

The design of a propeller is quite a demanding task because several conflicting factors play at the same time, such as: the engine power and shaft speed matching, the boat size and the ships operating speed. Each propeller has several geometric characteristics that describe it [1], namely: D, P/D, blades, hub, sense of rotation, RPM etc. A blade can be defined as a solid body interpolating a set of airfoils. Therefore, given

[1]Corresponding author; email: antonio.bacciaglia2@unibo.it
[2]email: alessandro.ceruti@unibo.it
[3]email: alfredo.liverani@unibo.it

a generic propeller section, a lot of parameters are needed to describe its shape, such as: mean-line type, chord, camber, thickness and so on.

Propeller series design approach

Two different ways are available for the designer to design and optimize a screw properly. Naval architects can rely on theoretical propeller design methods (lifting-line/surface theories) using a computer software without geometry constraints or, as an alternative, they can use off-the-shelf series labelled propellers [2]. Despite its rough approximation (a specific propeller design is carried out by comparing it with a set of tabulated data for a set of representative shapes), the second method, namely the propeller series, is still valuable nowadays even if a large computational power is available to designers. It is widely used in the preliminary design of light or moderately loaded propellers and for those designers who cannot afford lifting surface software. Just to provide an example, among the available propeller series, the B-series is one of the commonly used, developed in the Netherlands Ship Model Basin. Design charts are available for B-series propellers, and a new design can be carried out by selecting the propeller whose performances match in the best way the application under design.

To support the previous statement, the source [1] describes a simple example of propeller optimization, based on B-series. Performance related data are given as curves for an assortment of propellers by varying the load in the form of pitch. If the optimum advance ratio and thrust coefficient are known, the designer can detect from charts the propeller available in a series assuring the best efficiency. It is worth noting that this process does not require software computation, but it relies on experimental performance charts and diagrams.

In the same article, the cavitation is deeply described: it can affect the propeller performances and it has a strong impact on the design process. This physical phenomenon, also known as "fluid vaporization", can be described as a phase variation observable in high speed fluxes when in a liquid the local absolute pressure equals the vapour ones at atmospheric ambient temperature. It can cause damages and blade erosion, thrust and propulsive efficiency reduction, noise at high frequency on board and in the surrounding environment, engine shaft vibrations and high local structural loads. To understand if cavitation occurs, it is necessary to evaluate the σ_N parameter, widely known as cavitation number and to compare it with the minimum pressure coefficient ($C_{p_{min}}$). In general, σ_N measures the tendency of fluid to cavitate: larger the value of σ_N, smaller the likelihood for cavitation since the criterion for cavitation inception is $-C_{p_{min}} > \sigma_N$ [2]. [3] proposes an interesting optimization method to efficiently design propeller taking into consideration cavitation analysis. Different constrains are used, as cavity area and volume velocity harmonics to avoid this highly dissipative phenomenon; [4] uses a similar approach to optimize a propeller in a uniform and non-uniform flow and compares them with a propeller designed via an existing lifting-line approach.

The paper [5] is another good example of development and implementation of an optimization procedure for a marine propeller based on systematic series. For any B-series screw, K_T and K_Q can be expressed as functions of Z, A_E/A_O, P/D, and J (see symbols meaning in the nomenclature section). Following the approach developed by these authors, the propeller design process is carried out by setting an objective function (i.e. η), subjected to different constrains as thrust required, structural strength and cavitation constraints. Moreover, input data (the design variables) must verify the boundary conditions in terms of available range. The source [6] uses a similar approach, but in this case the objective function is a weighted sum (where weights are set by the user) of propeller efficiency and vibrations.

Numerical-based design approaches

Thanks to the huge computational power available nowadays, it is possible to optimize a propeller using some codes and algorithms based on lifting-line methods or surface theories, without using the geometry constrains of the series propellers. These systematic series are based on a parent model with a constant pitch distribution towards the tip and a low skew angle [7]. It's interesting to note that modern propeller designs, with a reduced tip loading and higher skew angles, achieve a similar level of pressure distribution as the B-Series propellers but with up to 3% higher efficiency. This is to say that the propeller series approach may not give the optimized shape even if the design procedure is perfectly followed. The reason why software codes are needed to implement theoretical methods is that, compared to the design with a propeller series, the required inputs are more demanding since thrust and profile section drag coefficients must be known. With such methods, the designer numerically creates a series of propellers to find the most suitable preliminary design, which can act as the starting point for the analysis and optimisation phase.

Another example of propeller optimization procedure can be find in [8], based on theoretical methods. Given an existing propeller geometry, with some problems in terms of pressure fluctuations on the hull and noise, the author tries to optimize it, maintaining the same K_T, using a wide chord tip; this is done implementing an algorithm (KPA4 & KPD4 programs) based on vortex-lattice methods (VLM) to evaluate the propeller performances. VLM is a widely used method that provides in a precise way the open water characteristics, cavitation analysis, thrust and torque fluctuations for conditions quite close to the designed point. On the other hand, there are some uncertainties when J is close to 0 or when the pitch angle decreases considerably with respect to the design value; in this case, a production of reverse thrust can be noticed in the outer blade portion while the inner ones generate forward thrust: this is a kind of behaviour that the VLM method can hardly model and predict. In this case, the RANS (Reynolds Average Navier-Stokes) method could be used, with higher computational demand, but with better analysis performances. In literature there are contributions regarding propeller optimization examples using high fidelity approaches as Boundary Elements Method (BEM) and viscous solvers implementing RANS model with respect to usual lifting-line/lifting-surface tools. [9] presents a multi-objective optimization method in order to improve propeller efficiency, reduce cavitation and maximize ship speed, taking under consideration engine-propeller matching, using a combination of BEM and viscous solver.

Fixed vs Controllable pitch propeller

When the attention is focused on functional issues, propellers can be divided into two different categories: the fixed pitch propellers (FPP), that are considered more reliable and easy to install, and the controllable pitch propellers (CPP). Even if the FPP geometry is quite complex, its manufacturing process is easier compared to CPP because of the lack of movable and settable parts. Maintenance and consumption are very low too. These positive aspects, together with good efficiency values in the design condition (the propeller is optimized for a defined advance velocity value), make FPPs widely used in general purpose marine application. The boat speed change is obtained varying the propeller rotational speed, increasing the engine throttle (consequently power and RPM). The propeller skew is introduced to decrease the load variations when blades, during operations, go through a weak peak, trying to decrease the cavitation inception risk. In general, a fixed pitch propeller is optimized for a single condition, or in other words, at a certain advance ratio and revolution per minutes that corresponds to the so called "design condition". In that operative circumstance, the efficiency is maximum, while for all the other RPM (off-design condition) the performances decrease dramatically. If a boat operates through a wide range of speed, a fixed pitch propeller can be inadequate due to a plunge of efficiency for velocities different from the designed one.

On the other hand, CPPs are more suitable to increase manoeuvrability, especially for sudden stops, or to increase efficiency when a ship has been designed to operate in a wide range of velocities and performances have to be optimized. Another difference is that the CPP hub ($D_{hub} = 0.28 \div 0.32D$) is bigger with respect to a FPP($D_{hub} = 0.15 \div 0.2D$), because it hosts a more complex, expensive and bigger mechanism for pitch change. The hub dimension increase leads towards a decrease of the available blade area and, as a matter of fact, to a slightly efficiency decrease. The boat speed change is obtained using a variation of the pitch angle instead of RPM variation (as it is FPP), maintaining suitable propeller performances, and allowing the engine to operate where fuel consumption is lower. An example of CPP application for fast ferries is described in [10] where good performances and low fuel consumptions in harbour and cruise speed conditions are needed. Moreover, in [11], it is described an optimization method based on coupling of multi-objective optimization algorithm and a panel code applied on a CPP mainly focusing on cavitation and consequently noise reduction at low RPM ranges.

Bibliography describes applications of controllable pitch propeller. Just to provide an example, [12] presents the ship speed spectrum records in open-sea operations: it shows a wide range of speed and pitch angle setting, suggesting CPP necessity for the specific usage investigated in the study by using the Vortex-lattice method to predict performances.

Scope of the work

It is worth noting that modern marine engineering is more and more focused on fast ships, as [13] suggests, where the design of a foil based yacht is described. When dealing with fast boats, one of the possible ways to obtain high propeller efficiency along the whole speed range envelope relies on the adoption of variable pitch propeller. Therefore, the aim of this paper is to extend the design methodologies based on the optimization of propeller shapes conceived for FPP to CPP models. As extensively presented in this introduction, some articles cited above deal with the design by optimization of marine propellers, making use of higher fidelity approaches (BEM and RANS) compared to usual lifting line/lifting surface tools. However, due to the high numbers of simulations to carry out with heuristic methods, and the need for short simulation times a simpler model has been implemented, since this kind of optimization should be used in the conceptual/preliminarily design phases to evaluate a lot of possible scenarios and for trade-off analysis, even if more detailed design approaches can be used in the following detailed design, which is beyond the scope of the paper.

The design method presented in this research, based on the OpenProp code's integration in the optimization loop, can be seen as an alternative to the serial propeller design approach, where the user selects the most suitable off-the-shelf products. In this paper, a more end-application-oriented approach is followed and a customized approach is proposed to design a controllable pitch propeller for small leisure ship boats. In agreement with [5], an objective function aiming to increase efficiency (and thus to reduce fuel consumption) and constrains (which allows to obtain a feasible solution) are imposed to search the optimum design. In particular, the optimization process is based on the Particle Swarm Optimization algorithm, which is used to drive the geometrical design of a CPP towards an optimal solution. The final result is a propeller whose efficiency is quite un-sensitive to the advance velocity, so that an almost uniform efficiency is guaranteed at low, medium and high speed. With respect the available literature, in this work, a different approach has been followed compared to literature since the propeller design has been carried out considering the engine power and consumption performances for different round per minute and throttle settings. Moreover, also a simplified model of ship drag is considered into the optimization loop to obtain a ship/engine/propeller proper integration.

The paper is structured as follows. After this introduction, a section describing the PSO algorithm is included. A section describing the methodology implemented to optimize the propeller follows. In the following, a case study where the methodology is applied to a real case of design is presented. Finally, a conclusion part is included where the results obtained are commented and an evaluation of the methodology is carried out.

2. Design tools

This section provides a brief description of the tool embedded in the optimization procedure implemented and tested in this paper. As explained in details in the next section, OpenProp open code for MATLAB has been exploited to compute the blade performances, while Particle Swarm Optimization algorithm drives the optimization towards the goal set by the designer.

2.1. OpenProp code

OpenProp [14] is one of the most valuable open-source software that implements the VLM: it is written in MATLAB environment and available in the web. This tool is based on lifting line theory, where each blade is seen as a lifting line with trailing vortices aligned to local flow. The induced velocity components, that contribute to the overall total inflow, are computed using the standard propeller vortex lattice model [15] with helical trailing vortex shed filament at discrete stations along the blade. As better detailed in the next section, the optimization framework developed in this paper allows the design of a controllable pitch propeller with given input data about the thrust requested by a ship along its operational speed range, and torque-RPM-throttle curves for motor. The reference [16] explains the matching relationships between engine power and propeller torque to be considered during an optimization process. In detail, ship propulsion is obtained through the thermal energy conversion, obtained from fuel burning into thrust conversion. The basic configuration of a boat propulsion system consists of three major units: engine, transmission and propeller. These three major subsystems have to work together in an efficient and coherent way to be able to propel a ship at certain design speed. The main propeller task is to generate a force so that the Thrust Horsepower (THP) could overcome the Effective horsepower (EHP), that is the power required to tow a hull. THP and EHP do not have to match perfectly, but a certain margin has to be considered to take into account the propeller effect and its hydrodynamic drag, known as the deduction factor. As [2] says, ship resistance has different contributions, where the frictional one is the major contribution and it is generated from tangential fluid forces, or in other words, due to the fluid viscosity. Another important amount is given by the wave-making resistance and reflects the energy required to push the water out of the way of the hull, i.e. quantifying the energy spent that goes into creating the wave. Due to the phenomenon complexity, wave-making drag can not be easily theoretically modelled and it is always determined in an experimental way. Both frictional and wave resistance have different importance depending on the Froude number (Fr), which is defined as the square root of the ratio between the inertial force and the weight. In addition, some small contributions, known as residuary resistance are present: these are eddy resistance, viscous pressure drag, separation and wave-breaking resistance. It is worth noting that a model used for optimization can not be too detailed because it would take days to run simulations. More detailed design approaches can be used in the following detailed design, which is beyond the scope of the paper. This is the reason why in this work, and in the presented case study, only a simplified resistance model will be considered, where the frictional component mainly contributes, taking in mind that the methodology behind the optimization algorithm could be more precise if experimental data were available to model the wave-making resistance, or in other words, the tool the authors developed can be refined each time the resistance curve versus the velocity is known from towing tank tests, from series or statistical data, from RANS calculations.

OpenProp is used to compute the torque and thrust by the propeller at different RPM and power from motor. OpenProp is based on theoretical background (VL method), optimal for marine propeller design that approximate the blade as a lifting line, using *Morgan correction* to the camber and ideal angle of attack required to construct the 3D blade from 2D lifting-line results. To describe properly the propeller geometry, some parameters as pitch, chord, camber, thickness, rake, skew are given as a function of the radius. Rake is the axial distance from the mid-chord point at the hub section and at the section of interest, while the skew is the tangential component of the angle formed on the propeller disk between a radial line going through the hub section mid-chord point and a radial line going through the mid-chord's section of interest [1]. In this software, as it happens in naval engineering field, propeller performances are given in a dimensionless form using K_T and K_Q for a given advance ratio J. As a result, the efficiency of the propeller operating behind the ship is

$$\eta = \frac{P_T}{P_D} = \frac{T\bar{V_a}}{2\pi nQ} = \frac{J}{2\pi}\frac{K_T}{K_Q}\frac{\bar{V_a}}{V_s} \tag{1}$$

OpenProp, using the VLM approach, assumes a horseshoe vortex filament with circulation $\Gamma(i)$ that surrounds the i-th panel with helical trailing vortex filament shed from its panel endpoints. From this assumption, the axial and tangential induced velocity components can be estimated using [17] formulae, imposing the alignment of wake with local flow. OpenProp's aim is to find the optimum propeller having an optimum circulation distribution, which means lower torque (and following power required by engine) at the desired thrust T_{des}. This optimization problem is solved iteratively using a Lagrangian multiplier λ_1 according to the equation:

$$H = Q + \lambda_1(T - T_{des}) \tag{2}$$

where the optimization process tries to find the H minimum. After the propeller optimization, the off-design conditions are computed with an iterative process (Newton solver) to find a set of unknown variables: resultant inflow velocity, circulation, ideal pitch angle, among others.

OpenProp has been widely validated, and it is a valuable tool to design a marine propeller with good performances, as it can be seen by the benchmarking activities held on the U.S. Navy propeller 4119 [18]: a good agreement between OpenProp and experimental tests for both design and off-design condition has been shown. However, there are some limitations related to the use of such a software: the main one is that Open Prop does not take into account the presence of a non-zero skew and rake angle on propeller performances, even if these setting are widely used in naval to reduce cavitation and noise. Moreover, as it happens for all the software packages that implements a vortex lattice method, there are some problems when the advance ratio becomes small or when the pitch angle decreases too much. In such conditions, there is a production of reverse thrust in the outer blade portion, while the inner one generates a positive contribution towards the direction of motion. In this context, the RANSE or BEM method could be used, with higher computational demand, but with better analysis performances. Nevertheless, the aim of this paper is to develop an easy and fast engineering tool useful to compute and 3D model controllable pitch propellers in the conceptual design phase and OpenProp matches the need for a reliable software to evaluate propeller performances, easy to embed into optimization codes and customize, and providing results in short times. Due to these capabilities, OpenProp macros will be exploited in the optimization process.

2.2. Particle swarm algorithm

The optimization methodology selected for this research is based on upon meta-heuristic algorithms. There are several ways to face an optimization process, and different kinds of methodologies can be adopted. Gradient based methods are useful to explore the design solutions space where the function to optimize is relatively simple and doesn't include local minimums. In such a scenario, the solution can be found following directions where the fitness function's derivatives increases or decreases [19]. Operative research based methodologies require a detailed knowledge of the model to optimize, and this does not apply in several design problems where simple linear models are not enough detailed to model phenomena in proper way. On the other hand, Monte Carlo [20] purely random methodologies do not require a detailed knowledge of the phenomenon to optimize, which can be considered as a "black box" where given a set of input, an output is obtained. However, Monte Carlo method is not time efficient and it requires a lot of simulations to obtain a reliable result.

Meta heuristic algorithms can be considered as a trade off between a purely random approach, and more mathematically sophisticated methodologies. This kind of algorithms imitate the way in which nature operates or physic works. The behaviour of populations following Darwin's evolution law is modelled in Genetic Algorithms [21]. Just to provide a few by representative techniques, the spreading of sparks is imitated in Fireworks algorithms [22]. The way in which kingdoms and countries evolute in history inspired the Imperialist Competitive Algorithm [23]. The simulated annealing algorithm tries to formalize in a mathematical way how a metal particle behaves once cast metal cools [24]. Ant colonies algorithm [25] is an example of procedure where ant search for food strategy has been imitated to find a proper solution in an optimization problem. Non-dominated Sorting Genetic Algorithm (NSGA) [26] improves the adaptive fit of candidate population solutions to a Pareto front constrained by a set of objective functions and an application in the naval field is presented in [27].

Among the bulk of meta heuristic strategies, Particle Swarm Optimization (PSO) algorithm assures the achievement of good results with a simple and straightforward code. PSO has been developed by Kennedy in 1995 [28] trying to imitate the dynamics of flocks of birds or school of fish while searching for food. It has been then applied in a wide set of case studies, ranging from aerospace engineering [29] to alternative energies [30], naval [31] and mechanical engineering. The PSO algorithm works assimilating the position of a possible solution in the design space where several parameters define a single solution, as a bird/fish in the n-dimensional space trying to find food. The position aimed by the individual during the space exploration depends on a combination of the best place found by the single element of the population and the best position found by the entire flock/school. From a mathematical point of view, an iterative update of position and speed is obtained using the eq. (3) and (4).

The particle speed can be expressed with

$$v_i(k+1) = \varphi(h)v_i(k) + \alpha_1[\gamma_{1i}(p_i - x_i(k)] + \alpha_2[\gamma_{2i}(G - x_i(k)] \tag{3}$$

where i stands for the index of the single particle, k expresses the algorithm step, $\varphi(k)$ expresses the inertia function, v_i symbol relates to the velocity of the i-th particle, $\alpha_{1,2}$ are acceleration constants, p is the best position found by the i-th particle (the so called personal best), G is the best position found by the whole swarm (best position within the personal bests), and finally $\gamma_{1,2}$ is a random number in the interval [0÷1].

The eq. (4) expresses the position's update of the particles:

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{4}$$

where x_i is the position of the i-th particle, and $v_i(k+1)$ is the updated velocity of the i-th particle. The simulation's termination can be achieved in two ways: in the first one the algorithm ends after a pre-set number of iterations. A second exit strategy can be reached when no improvement in solution is achieved for a pre-set number of consecutive iterations. The simulation ends when all the particles collapse into a single point in the n-dimensional space, the optimal one. In the first step the velocity is null for all the particles, and the position of the particles is random within the allowable parameters proper range.

3. Design process

In this section, the optimization process will be described in details. At first, the objective function $G(X)$ to minimize has been defined based on the designer's aim: it describes the minimization of the mean fuel consumption (FC) along three (low, intermediate and maximum) design velocities (respectively V_1, V_2, V_3). FC is obtained as the specific fuel consumption in $kg/(W \cdot h)$ times the generated power, since this quantity is not constant and depends on the mounted propeller. To simulate a boat velocity spectrum not homogeneous in order to define an operative profile of the ship, some weights (a_1, a_2 *and* a_3 respectively for V_1, V_2 *and* V_3) are used to consider not equally weighted functioning conditions and can be easily changed by the designer, in order to implement a more realistic optimization for the specific design-case (see eq. 6). This change directly affects the simulation output, since the fitness function is obtained by a weighted mean of the fuel consumption. The input vector X is defined in order to contain all the parameters necessary to define the propeller's shape and features:

$$\mathbf{X} = \{RPM, V_s, \beta_i, P/D, c/D, t/c, f/c, skew, rake\} \tag{5}$$

The algorithm aim is to find an optimal configuration, or in other words an optimal input vector which will be identified in the following as X_{opt}, that minimizes $G(X)$.

Some constrains limit the possible configurations in terms of cavitation occurrence, material structural strength and minimum generated thrust. Therefore, the resulting propeller has to work properly, without cavitation problems, far from the material yield stress and obviously it must provide enough thrust to move the boat at the desired speed. In order to satisfy these constrains, the designer has to know preliminarily data regarding the hull geometry and its hydrodynamic properties, the candidate material to be used to manufacture the blade, and finally the engine characteristics. These data are required to gather an optimum matching between ship, propeller and engine. The diameter can be unknown in a preliminary/conceptual design phase, but it is not a free variable since there are geometrical and physical constraints (pressure pulses on the hull stern or clearance to hull surface, for instance) that limit the propeller dimensions. The setting of allowable ranges for design parameters is one of the main problems in optimizations. However, some engine suppliers suggest an optimal diameter to obtain a good matching with that engine. Indeed, since this tool is developed for small leisure ship boats, it can be assumed that the final propeller diameter could be at around the diameter suggested by manufacturers, and a matter or fact its value can be used as a starting point for the optimization process. However, in case of commercial or navy ships the most important constraint on diameter is the shape of the stern and only after the selection of the propeller the appropriate engine is chosen.

It is worth noting that the code herein described is particularly suitable in the conceptual design phase since several design scenarios can be explored in days, thus providing the designer with a lot of what-if case studies. As a consequence, the mathematical problem to solve can be expressed as:

$$\begin{cases} \text{Find } \mathbf{X} \mid \textbf{min } G(X) = \frac{a_1 \cdot FC_1(X) + a_2 \cdot FC_2(X) + a_3 \cdot FC_3(X)}{\sum_{i=1}^{3} a_i} \cdot k_{cav} \cdot k_{stress} \cdot k_{thrust} \\ \textbf{Satisfying the constrains:} \\ \text{stress}(X) \leq \text{Max load} \\ \frac{-C_{p_{min}}}{\sigma_N(X)} \leq 1 \\ T_{des}(X) \geq \frac{Drag}{(1-t)} \end{cases} \tag{6}$$

To drive the Particle Swarm Optimization algorithm towards feasible solutions, some penalization coefficients (k_{cav}, k_{stress} & k_{thrust}) multiply the objective function. In particular, if the algorithm outputs a configuration that satisfies all the constrains in eq. (6), all the penalization coefficients will be equal to one. On the other hand, if one or more of the constrains are not respected, the corresponding coefficient penalizes the objective function (with a value > 1), increasing FC and so making the propeller configuration not-optimal. At each iteration, each penalization coefficient value is a weighted combination of the values at each designed velocity, depending on the required velocity spectrum, i.e. the a_i values. Penalization coefficients depend on the distance from the allowable solution, so that the PSO algorithm can feel the direction of improvement of the final solution in an easy way. The penalization coefficient values have been expressed through some lookup tables which can be customized by the user. The shape of the penalization coefficient functions impacts the convergence time of the solution because it affects the PSO algorithm capability of pointing towards a feasible solution. The PSO algorithm should sense the direction where fitness function improves in a continuous way: a step penalization function would be unsuitable because it wouldn't provide information about changes useful to improve solution. The penalization function shape isn't linear in order to move from design space zones where constraints are far from being satisfied. Just to provide the reader with an example, the tables for the case study which will be presented in the next section are shown in Figure 1.

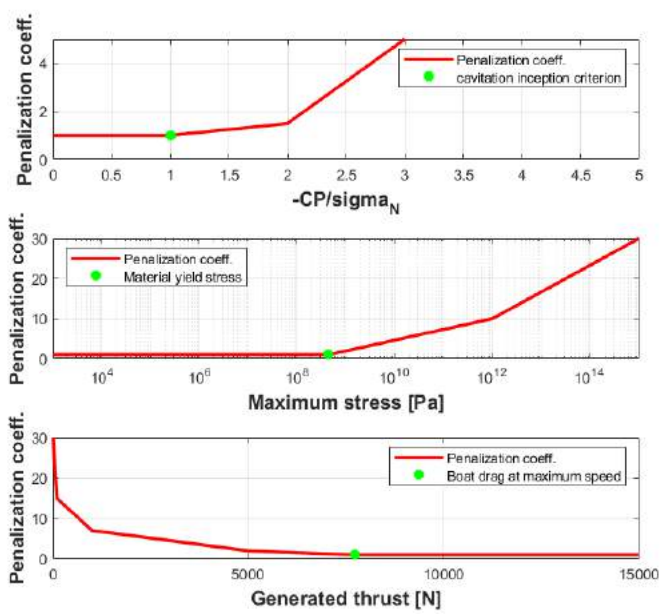


Figure 1: Penalization coefficients behaviour

The whole optimization methodology is explained in its implementation taking as a reference Figure 2, where each step of the procedure is labelled with a number from 1 to 18. Before starting the simulation, the algorithm has to be initialized; to do that the range of the parameters' variability is centred around the values corresponding to the reference propeller geometry, proposed by OpenProp software [18] when the medium velocity is set. The geometric parameters computed by OpenProp with such configuration are a good starting point for the optimization process. In case of a new design where no historical data are available, a diameter range of 20% around the value suggested from the engine supplier, can be set by the designer. This can support the experimenter in setting proper ranges for design parameters. Moreover, this helps the algorithm avoiding solutions difficult to analyse computationally. The number of blades can be changed by the experimenter, but in the case study presented in this paper it has been fixed to 5 (See Figure 2, stage **(1)**). At each iteration, the input vector X is set by the PSO algorithm **(2)**, where each parameter is chosen within an allowable range, as shown in Table 1 where data related to the case study presented in this paper are included. Some geometrical characteristics, like pitch and chord, are described using values at root, tip and at 75% of the radius. Then, cubic interpolation curves have been exploited to find the geometrical parameters in a continuous way along the radius. In particular, the PSO algorithm, at each iteration, gives as an input all the geometrical characteristics at 75% of the radius, that, for better understanding, will be called as reference section. The corresponding characteristics at hub and tip will be given as a certain percentage of the reference one, where percentage amount is free to change along the algorithm iterations.

To complete the inputs needed for the optimization, engine maps available from literature can be used to get the engine power and FC curves as a function of RPM and throttle percentage ($P_D = f(RPM, \%_{thr})$ and $FC = f(RPM, \%_{thr})$).

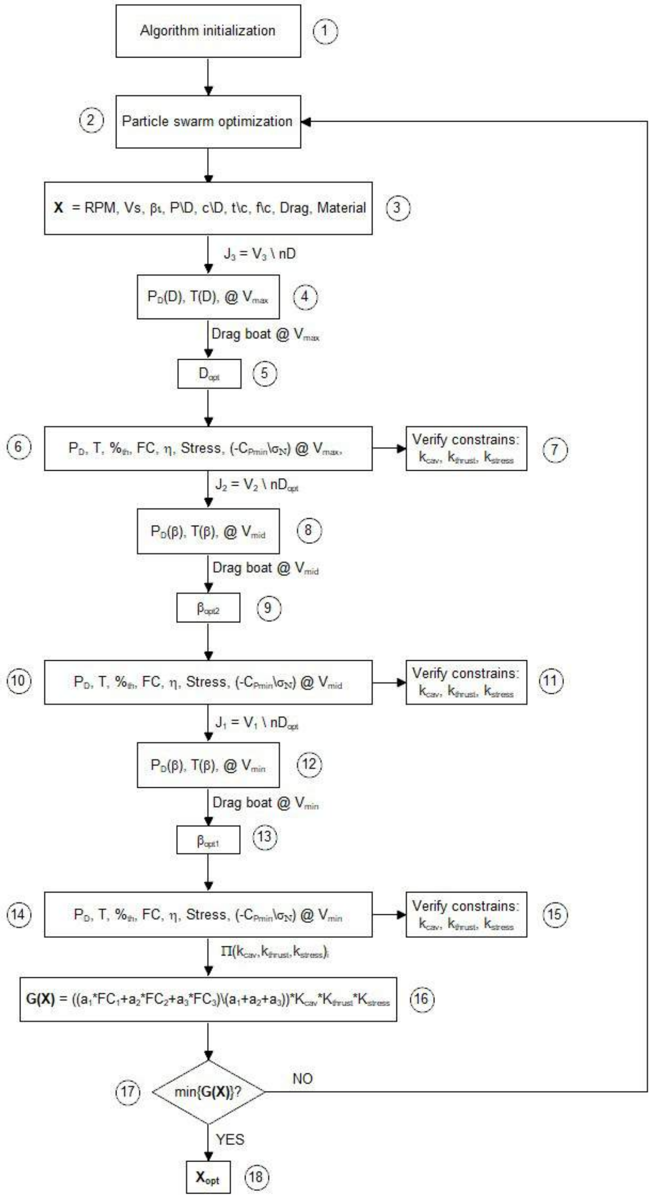


Figure 2: Design process flow chart

The optimization process starts analysing the propeller at maximum advance speed for a fixed RPM value, once the input vector X **(3)** is known. Using the OpenProp macro *AnalyzeGeometry*, the algorithm evaluates η, T, P_D, FC for different diameter values $D(i)$ referring to eq. (7).

$$\begin{cases} T(D(i)) = K_T \rho n^2 D(i)^4 \\ P_D(D(i)) = 2\pi K_Q \rho n^3 D(i)^5 \end{cases} \tag{7}$$

From this analysis it is possible to obtain a $P_D(D)$ and $T(D)$ behaviour through a mathematical interpolation **(4)**, where the diameter is not completely free to change, but it lies in the recommended range. By computing the hull drag, that corresponds to the frictional one because of the reasons previously explained and considering conveniently a deduction factor $t = 0.15$, the optimum diameter (D_{opt}) can be evaluated **(5)** in order to obtain a thrust that equals boat drag at maximum speed, or in other words, the design condition (Figure 3). However, in case the more detailed data about the ship drag are available from towing tank

Symb.	Description	Range
D	Propeller diameter	$0.32 \div 0.48$
RPM	Propeller rotational velocity	$980 \div 1820$
β_i	Initial stagger angle	$2.5 \div 10$
$K_{\phi_{hub}}$	Scale factor for pitch angle at hub section with respect the one at 75% of the radius	$1.8 \div 2.3$
ϕ_{75}	Pitch angle at 75% of the radius section	$12.5 \div 50$
$K_{phi_{tip}}$	Scale factor for pitch angle at tip section with respect the one at 75% of the radius	$0.6 \div 0.8$
$K_{c_{hub}}$	Scale factor for chord length at hub section with respect the one at 75% of the radius	$0.3 \div 0.55$
c_{75}	Chord length at 75% of the radius	$0.155 \div 0.62$
$K_{c_{tip}}$	Scale factor for chord length at tip section with respect the one at 75% of the radius	$0.7 \div 0.9$
t_{hub}	Thickness over chord length at hub section	$0.103 \div 0.411$
t_{75}	Thickness over chord length at 75% of the radius section	$0.027 \div 0.108$
t_{tip}	Thickness over chord length at tip section	$0.016 \div 0.063$
f_{hub}	Camber over chord length at hub section	$0.007 \div 0.029$
f_{75}	Camber over chord length at 75% of the radius section	$0.01 \div 0.04$
f_{tip}	Camber over chord length at tip section	$0.006 \div 0.024$
s_{hub}	Skew at hub section	$0.00005 \div 0.0002$
s_{75}	Skew at 75% of the radius section	$0.00005 \div 0.0002$
s_{tip}	Skew at tip section	$0.00005 \div 0.0002$
$rake_{hub}$	Rake at hub section	$0.00005 \div 0.0002$
$rake_{75}$	Rake at 75% of the radius section	$0.00005 \div 0.0002$
$rake_{tip}$	Rake at tip section	$0.00005 \div 0.0002$

Table 1: Input vector variables range

tests, series or statistical data, RANS calculations, the more complex expressions for resistance can be easily implemented in the optimization loop and more accurate results can be obtained.

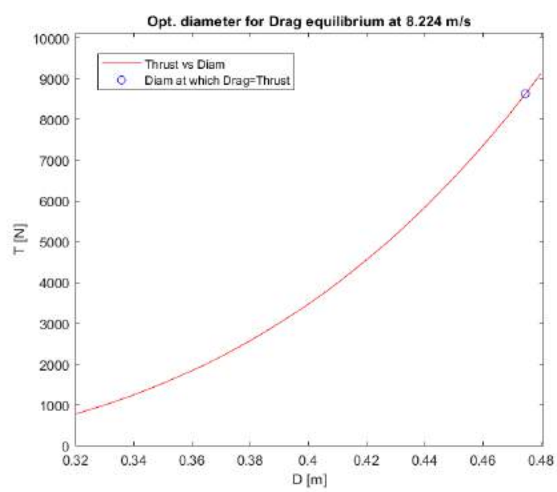


Figure 3: Optimum diameter given by condition of equilibrium

At this design stage, the optimal pitch distribution for the maximum speed is frozen as for the diameter, that will be fixed to D_{opt}, while for the other velocity conditions (medium and low) only the pitch angle will be changed to optimize propeller performances, as commonly done in CPP applications. Once the diameter has been set, the power absorbed by the propeller is available and, as a matter of fact, throttle % and consumption rate could be find at maximum speed for the given RPM value **(6)** thanks to the engine curves. To get suitable performances also at the other design conditions, the optimization approach, with OpenProp calling, looks for a radial pitch distribution that, combined with settling angles, provides optimal condition for the considered speed.

The second design step requires to find a settling angle β_2 combined with a pitch distribution for the mid advance speed to maintain reasonable efficiency values. Sweeping a large settling angle value range, the function evaluates again efficiency, power and thrust, obtaining a $P_D(\beta)$ and $T(\beta)$ curves through interpolation **(8)**. Therefore, the optimum value for settling angle (β_{2opt}) could be estimated from the equilibrium point, or in other words when the thrust developed is equal to the boat drag at Vs_2 **(9)**. When the optimal settling angle is available, the algorithm evaluates performances, throttle and consumptions for the mid advance velocity **(10)**.

As for β_{2opt}, the final step is to evaluate the combination of optimal settling angle β_{1opt} and radial pitch distribution for the lowest advance velocity value with the same procedure (**(12), (13), (14)**).

For each advance speed, the design constrains must be verified: hence, a structural analysis to evaluate the maximum stress level in the blade and a cavitation analysis are performed. This allows the optimized propeller to be durable, strong and performable. Two OpenProp macros (respectively *StressAnalysis* and *Cav_CavitationMap*) are used to this aim. In particular, *StressAnalysis* estimates the maximum stress given the propeller characteristics, V_s and RPM. The stress evaluation takes into account the centrifugal forces and the effects of forces and moments (including torque) due to the lift and drag on the blade. Additional moment contributions are due to the added mass forces and inertial accelerations. *Cav_CavitationMap* creates a cavitation map of the blade; σ_N is evaluated by knowing the pressure distribution along the blade. For additional information related to the mathematical model herein adopted the reader is addressed to the reference [14]. Afterwards, the design constrains are verified and the penalization coefficients are evaluated for each design velocity (**(7), (11), (15)**). Moreover, thanks to the engine map it is possible to match the propeller with a given engine throttle value. Figure 4 shows an example of engine map for different throttle settings and the matching point with the designed propeller performances of the proposed case study. It is worth noting that for a CP-propeller, the engine RPM will be fixed along the different design conditions, thanks to the controllable pitching mechanism: this allows the engine to work in a point where best FC can be achieved.

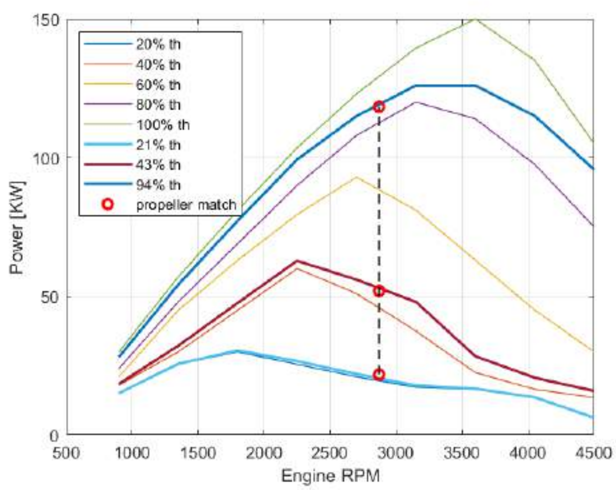


Figure 4: Engine map and optimum propeller matching, for the case study proposed in Section 4

At each iteration, the algorithm evaluates $G(X)$ **(16)** and, if the objective function is minimized, it saves the vector X **(17)**. When a new input array brings the objective function to a lower value, X is replaced with the new set of data and after the process is done, it's possible to know X_{opt} **(18)**. Using the optimum array data, the propeller performances along the three advance velocities are computed, using OpenProp capability in a precise, quick and easy way.

3.1. Controllable and fixed pitch propeller comparison

To assess the algorithm potentiality, a procedure useful to compare the changes in performances behaviour between a fixed pitch and controllable pitch propeller has been implemented. For the sake of comparisons with the CPP performances, three different FPP are designed to be efficient respectively at three operative velocity, V_1, V_2 and V_3. The FFP design can be easily the outcome of OpenProp used as a design tool, once the Diameter, the RPM and the required Speed/Ship Drag are defined at the advance velocity.

Thanks to this second procedure it is possible to draw performances chart for both FPP and CPP as a function of the advance velocity, so that the designer can be aware of the benefit obtained with CPP and cost-performances trade-off analysis can be carried out.

4. Case study

The case study used to validate the procedure deals with a propeller to be installed in a small leisure ship boat, which is designed to operate in a wide range of speed ($Vs = 7 \div 16kts$). Due to the drag evaluation assumption described in Section 2, the frictional resistance is assumed to be equal to the total one (neglecting all the other drag sources) and has been computed for this boat with the equation

$$Drag = \frac{1}{2}\rho S V_s^2 C_D$$

where a wet surface S, approximated as first attempt as the product of the length times the boat width of almost 15 m^2 has been assumed. In addition, it is assumed that the designed propeller has to be installed in a small leisure ship boat that works for 10% of its operations at low speed (V_1=7 kts), for 50% at intermediate speed (V_2=12 kts) and for 40% at the maximum speed (V_3=16 kts). The range of design parameters has been already shown in Table 1, while the function to be optimized tries to reduce as much as possible fuel consumption for V_1, V_2, V_3. Penalization functions limit stress to $440MPa$, avoids cavitation inception, and set to $T = \frac{Drag}{1-t}$ the lowest thrust value. The power-throttle-RPM characteristics of the engine have been included in Figure 4, where the maximum power of 150 KW is obtained at around 3600 engine RPM, similar to the D3 200 A Volvo Penta Aquamatic Duoprop™, available off-the-shelf. A gearbox with ratio 1:2.3 has been selected, being available on the shelf for this kind of motor. In fact, it is assumed to adopt commercial gearboxes which are available for marine engines, and usually sold with the engine.

The whole optimization has been run on a workstation with a 4core 3.1 GHz CPU and 32 GB of RAM and took 137 minutes to complete. The fitness function trend, along the optimization process can be seen in Figure 5, where after a fast drop, it stabilizes itself at the value $G(X_{opt}) = 22.6[kg/h]$. The optimal

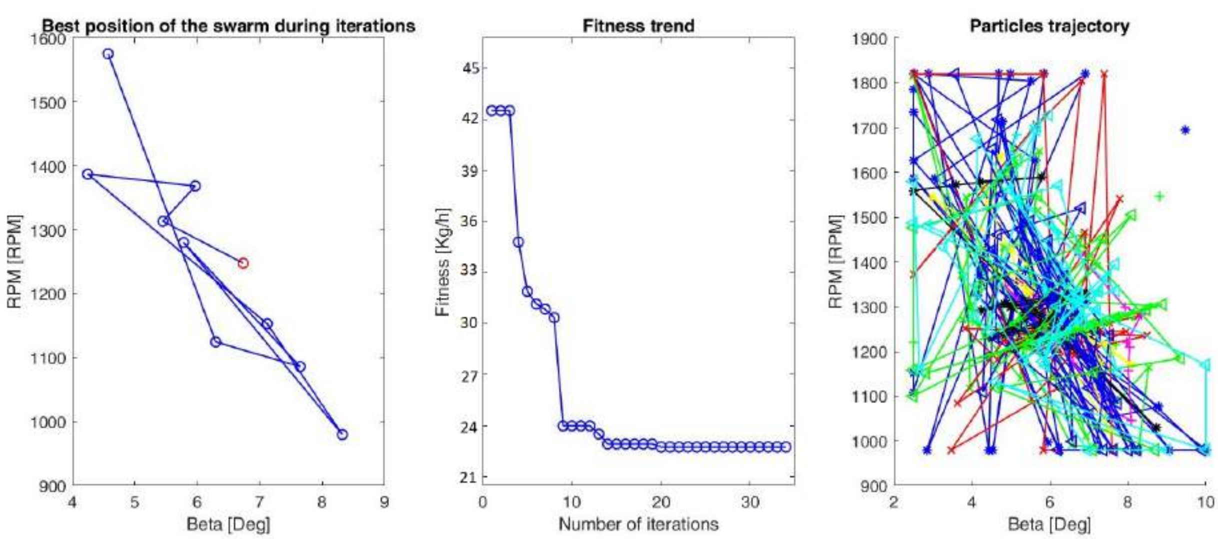


Figure 5: Objective function behaviour during the particle swarm optimization process, with particles' trajectories inside the design domain and the best $\beta_i(RPM)$ position

input vector, made of twenty different parameters, was selected by the particle swarm algorithm in order to minimize $G(X)$ where each resulting parameter is a certain percentage of the initial one. The algorithm stopping criteria is designed in a way that if the best fitness value does not decrease for 15 following algorithm iterations, the optimization process stops.

The resulting propeller, shown in Figure 6, satisfies all the constrains, making all the penalization coefficients equal to one. The resulting $G(X_{opt})$ value, physically acceptable, is the weighted mean consumption along the three design velocities.

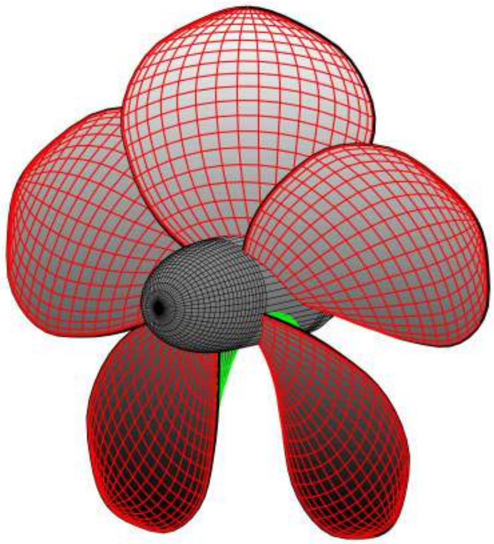


Figure 6: 3D view of the resulting controllable pitch propeller geometry given by the optimization algorithm

Propeller performances are collected in Table 2 according to the velocity profile previously described. It's important to underline that there is no throttle saturation even at the maximum velocity, the needed power is always lower than the available one, and the generated thrust is higher than the boat hydrodynamic drag.

	16 kts	**12 kts**	**7 kts**
Diameter $[m]$	0.473		
$\beta[deg]$	6.74	-3.4	-15.2
η	0.60	0.58	0.27
$P[W]$	118324	51975	21798
$T[N]$	8636	4883	1652
$\%_{th}$	93.8	42.9	20.7
$FC[kg/h]$	35.5	15.6	6.5
$stress_{max}[MPa]$	360	340	290
Cavitation inception margin	0.99	0.59	0.48

Table 2: CPP final performances

4.1. Global sensitivity analysis

After the optimization where a set of design parameters has been obtained using PSO algorithm, a sensitivity analysis has been carried out. For the specific case study, some additional numerical simulations were carried out to evaluate the impact of a design parameters change on the fitness function. To do that, one single parameter of X_{opt} is changed by 5% increase or -5% decrease, while the other inputs are kept constant. The fitness function is then evaluated. All the data are collected in Table 3. It is important to pinpoint that for each input parameter variation the fitness function increases: this is in agreement with the

Parameter	X_{opt}	$G(X_{opt})$	X_{opt} **+ 5%**	$G(X)$	X_{opt} **- 5%**	$G(X)$
D	0.473	22.65	0.497	34.89 (+54%)	0.449	38.19 (+68%)
RPM	1248.21	22.65	1310.62	39.63 (+75%)	1185.80	44.19 (+95%)
β_i	6.74	22.65	7.073	25.02 (+10%)	6.4	30.56 (+35%)
$K_{\phi_{hub}}$	2.072	22.65	2.176	23.38 (+3%)	1.968	31.47 (+39%)
ϕ_{75}	23.90	22.65	25.09	34.56 (+53%)	22.70	43.59 (+92%)
$K_{\phi_{tip}}$	0.798	22.65	0.838	25.75 (+14%)	0.759	37.62 (+66%)
$K_{c_{hub}}$	0.337	22.65	0.354	25.12 (+11%)	0.323	23.53 (+4%)
c_{75}	0.47	22.65	0.493	28.92 (+28%)	0.446	24.92 (+10%)
$K_{c_{tip}}$	0.888	22.65	0.932	22.75 (+0.5%)	0.843	25.74 (+14%)
t_{hub}	0.364	22.65	0.382	25.81 (+14%)	0.346	23.54 (+4%)
t_{75}	0.038	22.65	0.04	23.40 (+3%)	0.362	23.72 (+5%)
t_{tip}	0.017	22.65	0.018	24.13 (+7%)	0.016	22.99 (+1.5%)
f_{hub}	0.028	22.65	0.029	23.45 (+3.5%)	0.026	23.63 (+4%)
f_{75}	0.034	22.65	0.035	24.19 (+7%)	0.032	36.41 (+61%)
f_{tip}	0.024	22.65	0.025	23.61 (+4%)	0.022	24.12 (+6.5%)
s_{hub}	0.001	22.65	0.001	22.65 (+0%)	0.001	22.65 (+0%)
s_{75}	0.001	22.65	0.001	22.65 (+0%)	0.001	22.65 (+0%)
s_{tip}	0.001	22.65	0.001	22.65 (+0%)	0.001	22.65 (+0%)
$rake_{hub}$	0.001	22.65	0.001	22.65 (+0%)	0.001	22.65 (+0%)
$rake_{75}$	0.001	22.65	0.001	22.65 (+0%)	0.001	22.65 (+0%)
$rake_{tip}$	0.001	22.65	0.001	22.65 (+0%)	0.001	22.65 (+0%)

Table 3: Global sensitivity analysis: how changes in the inputs reflect on the objective function

final aim of the optimization algorithm whose task is the setting of design parameters values assuring the achievement of a minimum value fitness (which is the fuel consumption).

As it can be noticed, thanks to the global sensitivity analysis it is possible to demonstrate that the fitness function is always higher when the input array differs from X_{opt}, fact that it is fundamental in this kind of optimization process. As it may be expected, the more relevant parameter is the propeller RPM, followed by the diameter D and the pitch angle ϕ_{75} at 75% of the radius. Moreover, for the parameter which affect the fitness function whit higher impact (RPM), further sensitivity analysis is computed. The fitness function is evaluated when an increase or reduction of 1, 3 and 5% of RPM occurs, in order to understand if the optimum RPM value returns the lowest fuel consumption value, or in other words if the optimum set of design parameters is local or global. Figure 7 confirms that the set of design parameters values suggested by the PSO algorithm corresponds to a global optimum point in the fitness function.

4.2. Analysis of different velocity profiles

To understand how the algorithm behaves with different velocity profiles, some other scenarios have been simulated with different a_i weights. To understand reasons better, the simulated scenario proposed in the previous case study, which outputs are collected in Table 2, will be called *case a* (a_1, a_2 and a_3 respectively 0.1 0.5 0.4); a situation with the propeller equally used during a mission will be referred to *case b* (for $i = 1, 2, 3$ $a_i = 0.33$); *case c* (a_1, a_2 and a_3 respectively 0.05 0.8 0.15) and *case d* (a_1, a_2 and a_3 respectively 0.005 0.99 0.005) are used to simulate almost a FPP. Looking at the results, *case a*, for obvious reasons will have the highest fuel consumption because the ship will work for the 40 % of its mission at the maximum speed (Figure 8 [a]). If the reader refers to Table 2, it's clear that the efficiency is almost constant for the velocity profile portion that has highest influence. The results, coming from the propeller optimization when other velocity profiles are used, are collected in Table 4. Comparing the profiles a and b in terms of η and FC, it can be seen a more equilibrated trend for b, with higher efficiency for the lowest design velocity (Figure 8 [b]). Comparing the *case d* with all the previous ones, it can be seen that the algorithm focuses on the intermediate velocity that counts for 99% and, in order to minimize its fuel consumption, it neglects the FC for the other two conditions. For example, at the maximum speed, the thrust is slightly insufficient (96% of the drag at maximum speed which counts for 8635 N) and there is a small cavitation inception

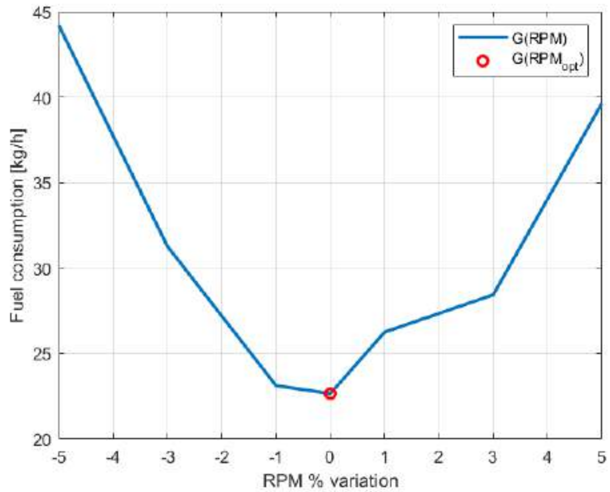


Figure 7: Objective function behaviour in case of changes on propeller RPM which is the most relevant input parameter

condition (the cavitation inception margin is higher than 1). As a consequence of the discussed results, this algorithm has good performances to get a first estimation in the preliminary design stage for controllable pitch propeller, while it demonstrate its limits for fixed pitch propellers.

	case b			**case c**			**case d**		
	V_3	V_2	V_1	V_3	V_2	V_1	V_3	V_2	V_1
a_i	0.33	0.33	0.33	0.05	0.8	0.15	0.005	0.99	0.005
$\beta[deg]$	3.30	-7.0	-12.40	4.36	-3.9	-13.3	4.42	-4.6	-16.3
η	0.61	0.57	0.37	0.61	0.56	0.30	0.60	0.58	0.35
$P[W]$	115595	49604	36333	117043	53510	19884	113788	53409	17190
$T[N]$	8643	4870	1561	8640	4856	1656	8279	5049	1679
$\%_{th}$	90.1	41.9	32.8	78.5	49.2	24.1	87.7	43.5	16.4
$FC[kg/h]$	34.7	14.9	10.9	35.1	16.1	5.9	34.1	16.0	5.16
$stress_{max}[MPa]$	370	330	280	480	470	330	390	350	230
Cav. margin	0.99	0.52	0.49	0.92	0.69	0.57	1.04	0.85	0.69

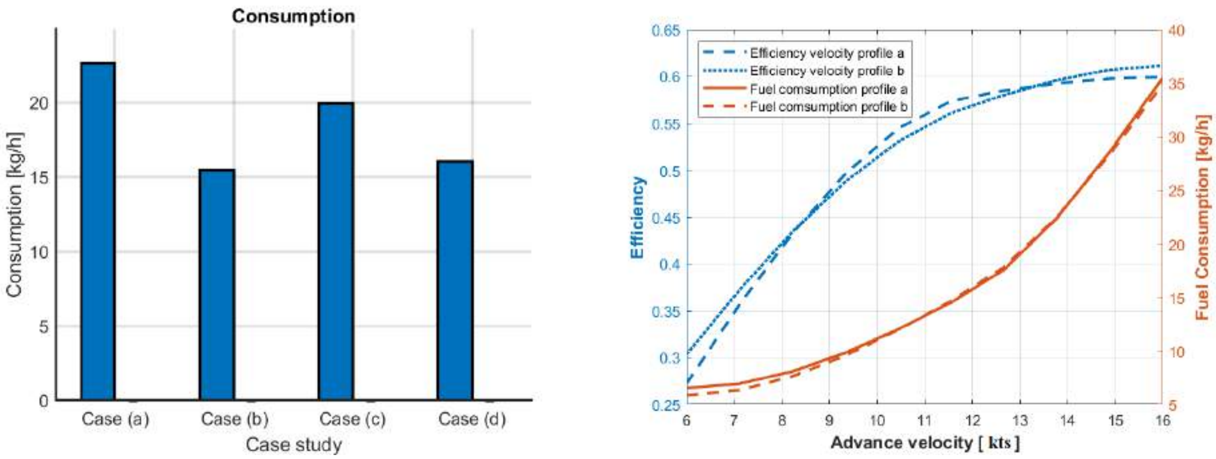
Table 4: CPP performances for different velocity profiles

4.3. Controllable pitch vs fixed pitch propeller

To describe the design process potentiality, a comparison between the CPP obtained by the Particle Swarm Algorithm and three FPP, each optimized at a single operative advance speed V_1, V_2 and V_3, is included.

In particular, from Figure 9, the reader can see η, K_T, K_Q and FC behaviour along a wide range of advance speed for all the four propellers under consideration. As commonly known, the fixed pitch propeller, Figure 9 [b], [c] and [d], works well in a small advance speed range because, moving far from the designed condition, it means a drastic drop of propeller performances in terms of efficiency and generated thrust. As a proof of that, the optimization code starts to diverge when the user asks to evaluate the propeller performances for advance speeds that are far from the optimum one. For example, the FPP optimized for V_3 has a negative η value for V_s lower than $10kts$, while the FPP optimized for V_1 has a FC that goes towards infinite when V_s decreases up to $0kts$. For this reason the Figures 9 [b], [c] and [d] are truncated along the x axis.

On one hand, a CPP, thanks to controllable pitch mechanism, can adapt itself to different operative conditions and as a matter of fact it can extend the useful advance ratio range $(5 \div 16)$ with good values in terms of propeller efficiency (Figure 9 [a]). Looking at the fuel consumption, the key parameter in this



(a) Fuel consumption comparison for different velocity profiles (b) Performance comparison for velocity profiles a and b

Figure 8: Performance comparison for different velocity profiles

case study, the CPP has a higher value at the maximum speed with respect to FPP, but a lower mean value considering a wide advance velocity range, making it preferable for applications where the vessel has to change quite often its operative speed. On the other hand, if the operative velocity is always almost the same, these results confirm that an FPP, properly designed to have η_{max} at the designed advance velocity, would be the right choice between the two options.

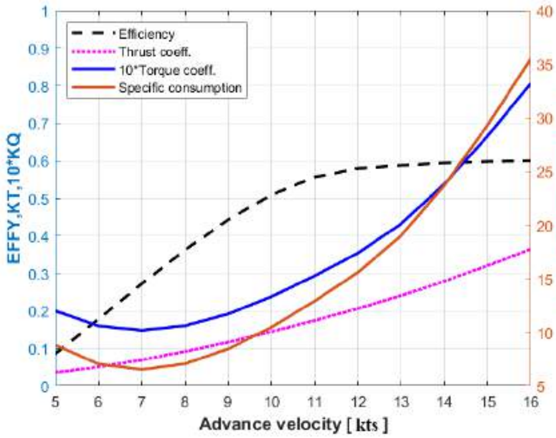
As a proof of the good performances assured by this algorithm, a blade of the optimized propeller has been manufactured using Additive Manufacturing technologies (AM) [32] to evaluate the final shape of the blade suggested by the procedure. A Form 2 Stereolithography (SLA) 3D printer [4] has been used to manufacture the part: it uses a laser to cure solid isotropic parts from a liquid photopolymer resin, delivering high-resolution parts (10). After a visual inspection the blade is similar to shapes typical of off-the-shelf CPP.

5. Conclusions

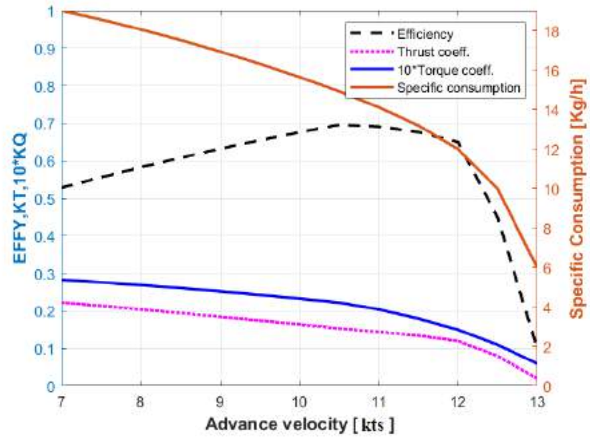
Modern marine engineering requires tailored products to obtain high performances and reduce the environmental impact of human activities. Increasing the efficiency of marine propellers leads to reduce fuel consumption and increase in overall efficiency. This paper presents a methodology to optimize controllable pitch propellers which could be used for small leisure ships. A procedure based on the integration of the open source OpenProp software into an optimization loop based on Particle Swarm Optimization semi-heuristic algorithm is described. Features of the engine selected and ship to power, range for geometric main parameters of the propeller are the main input of the optimization loop. The procedure gives as result, the blade geometry and the look up table pitch-speed which could be implemented in the electronics of the ship. Cavitation and structural issues are kept into account to obtain a solution feasible and ready to be manufactured. The efficiency of the controllable pitch propeller is less than what noticed for a fixed pitch propeller operating in the design point, but it is almost constant for a large range of velocities, thus allowing superior performances in a wide operational envelope. Future work includes the realization of a prototype of the whole controllable pitch propeller (controllable pitch mechanical system included) and a deep testing in water.

Funding and Conflict of Interest. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors also declare that they have no conflict of interest.

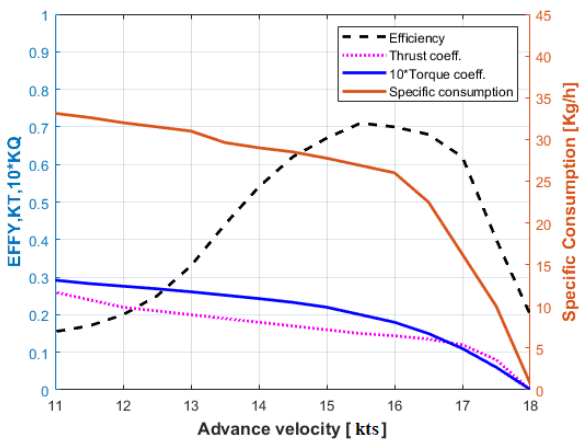
[4]Form 2 by Formlabs (https://formlabs.com/3d-printers/form-2/)



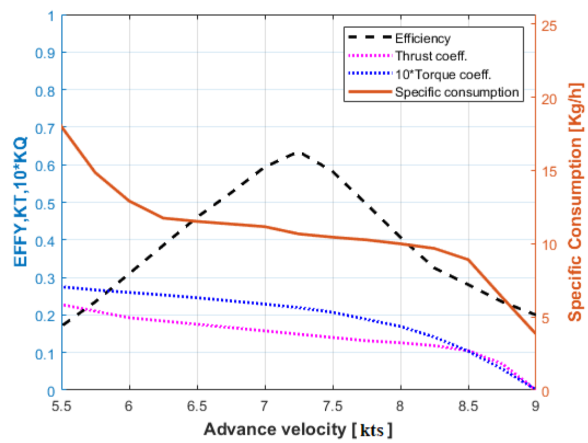
(a) Controllable pitch propeller performances



(b) Fixed pitch propeller performances optimized at V_2



(c) Fixed pitch propeller performances optimized at V_3



(d) Fixed pitch propeller performances optimized at V_1

Figure 9: CPP vs FPP performance plot

[1] A. Techet, Hydrodynamics for ocean engineers.

[2] E. V. Lewis, Principles of Naval Architecture - Volume II: Resistance, Propulsion and Vibration, The Society of Naval Architects and Marine Engineers, 1988.

[3] P. E. Griffin, S. A. Kinnas, A design method for high-speed propulsor blades, Journal of Fluids Engineering 120 (3) (1998) 556. doi:10.1115/1.2820698.

[4] S. Mishima, S. Kinnas, Application of a numerical optimization technique to the design of cavitating propellers in nonuniform flow, Journal of Ship Research 41 (2) (1997) 93–107.

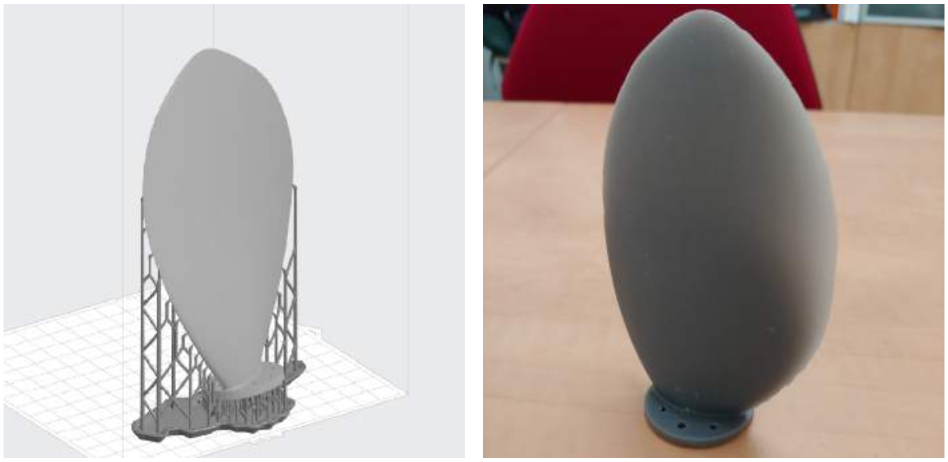
[5] M. Gaafary, H. El-Kilani, M. Moustafa, Optimum design of b-series marine propellers, Alexandria Engineering Journal 50 (1) (2011) 13 – 18. doi:https://doi.org/10.1016/j.aej.2011.01.001.
URL `http://www.sciencedirect.com/science/article/pii/S1110016811000093`

[6] J.-H. Chen, Y.-S. Shih, Basic design of a series propeller with vibration consideration by genetic algorithm, Journal of Marine Science and Technology 12 (3) (2007) 119–129. doi:10.1007/s00773-007-0249-6.
URL `https://doi.org/10.1007/s00773-007-0249-6`

[7] F. Vesting, Marine propeller optimisation - strategy and algorithm development, Ph.D. thesis, Chalmers University of Technology - Department of Shipping and Marine Technology (2015).

[8] C.-S. Lee, Y.-D. Choi, B.-K. Ahn, M.-S. Shin, H.-G. Jang, Performance optimization of marine propellers, International Journal of Naval Architecture and Ocean Engineering 2 (4) (2010) 211 – 216. doi:https://doi.org/10.2478/IJNAOE-2013-0038.
URL `http://www.sciencedirect.com/science/article/pii/S2092678216302503`

[9] S. Gaggero, G. Tani, D. Villa, M. Viviani, P. Ausonio, P. Travi, G. Bizzarri, F. Serra, Efficient and multi-objective cavitating propeller optimization: An application to a high-speed craft, Applied Ocean Research 64 (2017) 31–57. doi:10.1016/j.apor.2017.01.018.



(a) Controllable pitch propeller blade, coming from the optimization algorithm in Preform environment, ready to be manufactured

(b) Manufactured blade with rapid prototyping

Figure 10: Optimized blade designed and manufactured

[10] F. Balsamo, F. De Luca, C. Pensa, Continuous optimization of controllable pitch propellers for fast ferries, IX HSMV Naples 25 - 27 May 2011.

[11] D. Bertetta, S. Brizzolara, S. Gaggero, M. Viviani, L. Savio, CPP propeller cavitation and noise optimization at different pitches with panel code and validation by cavitation tunnel measurements, Ocean Engineering 53 (2012) 177–195. doi:10.1016/j.oceaneng.2012.06.026.

[12] T. Stoye, Propeller design and propulsion concepts for ship operation in off-design conditions, Second International Symposium on Marine Propulsors smp11, Hamburg, Germany, June 2011.

[13] C. L. Amoroso, A. Liverani, G. Caligiana, Numerical investigation on optimum trim envelope curve for high performance sailing yacht hulls, OCEAN ENGINEERING 163 (2018) 76 – 84.

[14] B. P. Epps, R. W. Kimballz, Unified rotor lifting line theory, Journal of Ship Research, Vol. 57, No. 4, December 2013doi:http://dx.doi.org/10.5957/JOSR.57.4.110040.

[15] J. Kerwin, Hydrofoils and propellers, 2007 (2007).

[16] D. Priyanta, Basic turbine - propeller matching, department of Marine Engineering ITS, Surabaya.

[17] J. Wrench, The calculation of propeller induction factors, Tech. Rep. 1116, David Taylor Model Basin.

[18] S. Black, Integrated lifting surface / navier-stokes design and analysis methods for marine propulsors, Ph.D. thesis, MIT (1997).

[19] V. Keshavarzzadeh, H. Meidani, D. Tortorelli, Gradient based design optimization under uncertainty via stochastic expansion methods, Computer Methods in Applied Mechanics and Engineering 306 (2016) 47–76.

[20] G. Fishman, Monte Carlo: concepts, algorithms, and applications., Springer, New York, 1995.

[21] D. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison-Wesley Pub. Co, Reading, 1989.

[22] Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)Volume 6145 LNCS, Issue PART 1, 2010, Pages 355-3641st International Conference on Advances in Swarm Intelligence, ICSI 2010; Beijing; China; 12 June 2010 through 15 June 2010;, 2010.

[23] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm for optimization inspired by imperialistic competition, in: S. IEEE Congress in Evolutionary Computation (Ed.), IEEE Congress in Evolutionary Computation, Singapore, 2007.

[24] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671680.

[25] L. M. Gambardella, M. Dorigo, Solving symmetric and asymmetric tsps by ant colonies, in: Proceedings of the IEEE Conference on Evolutionary Computation1996, Pages 622-627Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, ICEC'96; Nagoya, Jpn; ; 20 May 1996 through 22 May 1996, 1996. doi:http://dx.doi.org/10.5957/JOSR.57.4.110040.

[26] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation 2 (3) (1994) 221–248. doi:10.1162/evco.1994.2.3.221.

[27] J. Huismann, E. J. Foeth, Automated multi-objective optimization of ship propellers, Proceedings of the Fifth International Symposium on Marine Propulsors, SMP'17, Espoo, Finland.

[28] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks - Conference Proceedings, Volume 4, 1995, Pages 1942-1948 Proceedings of the 1995 IEEE International Conference on Neural Networks. Part 1 (of 6); Perth, Aust; 27 November 1995 through 1 December 1995, 1995.

[29] A. Ceruti, P. Marzocca, Heuristic optimization of bezier curves based trajectories for unconventional airships docking, Aircraft Engineering and Aerospace Technology 89 (1) (2017) 76–86.

[30] A. Ceruti, Meta-heuristic multidisciplinary design optimization of wind turbine blades obtained from circular pipes, Engineering with Computers 30 March 2018, Pages 1-17.

[31] F. Vesting, R. Gustafsson, R. E. Bensow, Development and application of optimisation algorithms for propeller design, Ship Technology Research 63 (1) (2016) 50–69. doi:10.1080/09377255.2016.1145916.

[32] A. Bacciaglia, A. Ceruti, A. Liverani, A systematic review of voxelization method in additive manufacturing, Mechanics & Industry 20 (6) (2019) 630. doi:10.1051/meca/2019058.