

## Article

# Direct Tracking of the Pareto Front of a Multi-Objective Optimization Problem

Daniele Peri <sup>†</sup> 

Istituto per le Applicazioni del Calcolo “Mauro Picone”, Consiglio Nazionale delle Ricerche (IAC-CNR),  
00185 Roma, Italy; d.peri@iac.cnr.it

<sup>†</sup> Current address: Via dei Taurini 19, 00185 Roma, Italy.

Received: 12 August 2020; Accepted: 7 September 2020; Published: 9 September 2020



**Abstract:** In this paper, some methodologies aimed at the identification of the Pareto front of a multi-objective optimization problem are presented and applied. Three different approaches are presented: local sampling, Pareto front resampling and Normal Boundary Intersection (NBI). A first approximation of the Pareto front is obtained by a regular sampling of the design space, and then the Pareto front is improved and enriched using the other two above mentioned techniques. A detailed Pareto front is obtained for an optimization problem where algebraic objective functions are applied, also in comparison with standard techniques. Encouraging results are also obtained for two different ship design problems. The use of the algebraic functions allows for a comparison with the real Pareto front, correctly detected. The variety of the ship design problems allows for a generalization of the applicability of the methodology.

**Keywords:** multi-objective optimization; pareto front; normal boundary intersection; ship design optimization

**MSC:** 78M50; 90C29; 58E17; 62K05; 49Q10; 46N10

## 1. Introduction

Most of the real-life optimization problems are multi-objective. It is quite uncommon that a single objective is sufficient for the determination of the optimal qualities of a design. As a consequence, tools for the determination and the analysis of the alternatives coming from a multi-objective design optimization problem are of great importance.

In the current literature, recently reviewed in [1], a very large number of optimization algorithms have been proposed to solve this task. Some of them are single-objective optimizers, applied to a single objective function where the different objective functions are recasted into a single objective function using some weights for the different objectives, being these weights static or dynamic. Another opportunity is the definition of a *goal programming* problem, where a specific value of each objective function is required, and the objective function to be minimized is represented by the Euclidean distance of the current solution and the target solution in the objective functions space. Other methods are adaptations of optimization algorithms formulated for a single objective: the different objectives are considered without an aggregation, and the concept of Pareto optimality is adopted. A typical example is the multi-objective version of the popular Genetic Algorithm (GA), namely Multi-Objective Genetic Algorithm (MOGA). The kernel of MOGA is exactly the same as GA, but only the fitness function, that is, the function providing the ranking of the different solutions, is different. Summarizing, none of the above algorithms are, in general, tracking directly the Pareto front (an exception is represented by the Normal Boundary Intersection method—NBI [2],) and the Pareto front is typically obtained by a recombination of the actual solutions.

In this paper, three different approaches aimed at the determination and enrichment of the resolution of the Pareto front are presented. A regular sampling of the Design Variables Space (DVS) is producing a first approximation of the Pareto front, and each successive step is producing an improved approximation of the Pareto front. Application to algebraic and industrial problems give positive indications about the efficiency of the approach.

## 2. Materials and Methods

The formulation of a multi-objective optimization problem requires the definition of *optimal point* in a multi-objective contest. In fact, it is absolutely uncommon to find a single solution that presents the minimum (or maximum) values for all the objectives at the same time. A widely accepted definition is the following:

**Definition:** The vector  $F(x)$  is said to dominate another vector  $F(y)$ ,  $x$  and  $y \in C$ , denoted  $F(x) \prec F(y)$ , if and only if  $f_i(x) \leq f_i(y)$  for all  $i \in \{1, 2, \dots, n\}$  (where  $n$  is the number of criteria) and  $f_j(x) < f_j(y)$  for at least one  $j \in \{1, 2, \dots, n\}$ . A point  $x \in C$  is said to be globally Pareto optimal or a globally efficient point if and only if there does not exist  $y \in C$  satisfying  $F(y) \prec F(x)$ .  $F(x)$  is then called globally non dominated or non inferior.

In this case, we have not a single optimal solution, but a variety of Pareto-optimal solutions, distributed in the so-called *Pareto front*, that is, the locus of the Pareto optimal points. The determination of the Pareto front represents the solution of a multi-objective optimization problem. It is an hard task, mainly because the Pareto front is defined in the objective-function space, and the relationship between the DVS and the objective-function space is typically not trivial. For a single-objective minimization problem we can easily find a search direction in the DVS, i.e., using the local gradient of the objective function, so we can move along this direction in order to detect an improved value of the objective function. This opportunity is not explicit for a multi-objective problem, because the definition of the Pareto front cannot be translated to the DVS. Furthermore, a single direction able to improve simultaneously all the objectives rarely exists: consequently, it is hard to determine a search direction driving us to a better Pareto point.

As already recalled, the algorithms for the solution of a multi-objective optimization problem are often adaptations of algorithms developed for a single-objective problem. One of the more representative examples is the classical approach for the adaptation of the GA to the Multi-Objective GA (MOGA) [3]. The Pareto front is evaluated using the current set of solutions, and the value of 1 is assigned to the fitness function for all the points belonging to the Pareto front. After that, a second level of the Pareto front can be determined excluding the previously detected Pareto points, and a value of 2 is assigned to the fitness function for these points (second level of the Pareto front). The procedure continues until a value of the fitness function has been assigned to every available solution. Since then, the typical operations of the GA are performed using a single valued objective function, as usual. In other words, the kernel of the algorithm is not changed, while the objective function is reformulated in such a way that the original multi-objective problem can be treated as a single-objective problem.

A more ambitious strategy could be to produce a search algorithm for the direct identification of new Pareto points, starting from a first approximation of the Pareto front. In this case, the search is not performed on the full DVS, but in selected regions where the new Pareto points are supposed to lie. We are here following three main guidelines:

1. **Local sampling**—Perform a search in a small region around each actual Pareto point.
2. **Interpolation**—Try to identify a curve in the DVS composed by the actual Pareto points and then compute new solutions refining and regularizing the resolution of this curve.
3. **Normal Boundary Intersection method (NBI)**—Identify a direction in the DVS, aligned with the local normal of the Pareto front, able to improve the current Pareto point.

Local sampling is very intuitive: we can find a true Pareto point in the neighbor (referring to the DVS) of an approximation of a Pareto set. We are hypothesizing a local continuity of the involved objective functions, and this hypothesis could be reasonable.

Interpolation is a little more reckless and simplistic: we cannot generally think that the continuity of the Pareto front represents a guarantee of a continuity for the Pareto points in the DVS. This statement is strongly different than the previous one, since we can have the same value of the objective function in different points in the DVS, not located side-by-side. Anyway, in this study we are going to test also this opportunity.

NBI is the much more solid strategy for the determination of new Pareto points. The original formulation can be found in [2]. The idea is that an approximation of the Pareto set can be improved if we are able to move the point along a direction normal to the Pareto front in the objective function space. This requires a sort of inverse map of the objective function, from the objective function space to the DVS, while we usually have the opposite.

More details for the different phases are provided hereafter.

### 2.1. Initial Sampling

A first approximation of the Pareto set is obtained by an uniform sampling of the DVS. Since we have not preliminary information about the location of the Pareto front, every point in the full DVS could be, at the start, a Pareto point. In other terms, the probability to find a Pareto point is uniform over the full DVS. As a consequence, in order to have a regular sampling of the DVS, we are using an Uniformly Distributed Sequence (UDS), that is, a point distribution in which all the outcomes are equally likely. Among the different strategies, we can use Sobol sequence [4], Latin Hypercube Sampling (LHS) [5] or D-Optimal design [6]. In this examples, we are using the Sobol sequence <sup>1</sup>.

### 2.2. Local Sampling

Once a first approximation of the Pareto front is available, a small UDS is placed around every Pareto point, performing a local search. A different Sobol distribution is generated for every point, and at each iteration a permutation of the design variables is applied, in order not to recompute the same points if the Pareto candidate is not improved.

### 2.3. Interpolation

The actual Pareto set is ordered in the DVS with respect to the first objective function, and then a curve passing thru these points is traced in the DVS. A gaussian filter is also applied in order to regularize the curve, since the resulting points are possibly poorly aligned. The curve is divided into a number of regularly spaced intervals, and the resulting points are computed.

### 2.4. Normal Boundary Intersercion—NBI

In the original formulation of NBI, the new tentative Pareto points are obtained by the solution of a suite of *goal programming* sub-problems, where the starting point are the actual approximations of the Pareto front and each with a constraint forcing the solution along a vector normal to the Pareto front. As a consequence, we are looking for an improvement of each point of the current approximation of the Pareto front, but along a precise direction. This could represent a limitation, since the real Pareto point could lie in the vicinity of the starting point, but in a direction, in the objective function space, not coincident with the local normal. Also for this reason, in this particular implementation we are not computing the normal to the Pareto front, and the new point is obtained, also in this case, by solving a *goal programming* problem. The target point is represented by a point in the objective function space whose values of the objective function are selected by following two different criteria<sup>2</sup>:

<sup>1</sup> the related software can be actually found in <https://people.sc.fsu.edu/~jburkardt/src/sobol/sobol.html>.

<sup>2</sup> we are considering here a minimization problem for all the objectives: in case of a maximization, indications should be reversed.

1. **Direct tracking:** the target point is defined by **decreasing** the objective function values of the actual Pareto point by a small amount, say 1%.
2. **Indirect tracking:** the target point is defined by **increasing** the objective function values of the actual Pareto point by a small amount. The new point is then mirrored with respect to the reference Pareto point in order to have an improvement for all the objectives.

Direct tracking is similar to the original NBI formulation with the exclusion of the constraint of the solution to stay along the normal of the Pareto front. The indirect tracking has been introduced because we are not sure about the existence of the point generated for the direct tracking option, for which the objectives are all improved. On the contrary, it is more reasonable to assume the existence of a point deteriorating all the objectives. The mirroring operation is performed hypothesizing a linear behavior of all the objective functions in a small neighborhood of the point, so that the deterioration along one direction indicates an improvement on the opposite direction: it can be reasonably applied only for small variations of the objective functions.

The solution of these *goal programming* problems could be very expensive, depending on the complexity of the objectives: if we are managing CPU hungry objective functions, this phase could take too long. For this reason, a multidimensional spline [7] is adopted in order to produce an algebraic approximation/interpolation of the objective function to be applied during NBI. The actual evaluations of the objective function are used as training set for the metamodel, so that the NBI takes a moderate amount of time. The solution of each sub-problem is finally computed by using the true expression of the objective functions.

### 3. Algebraic Test Functions

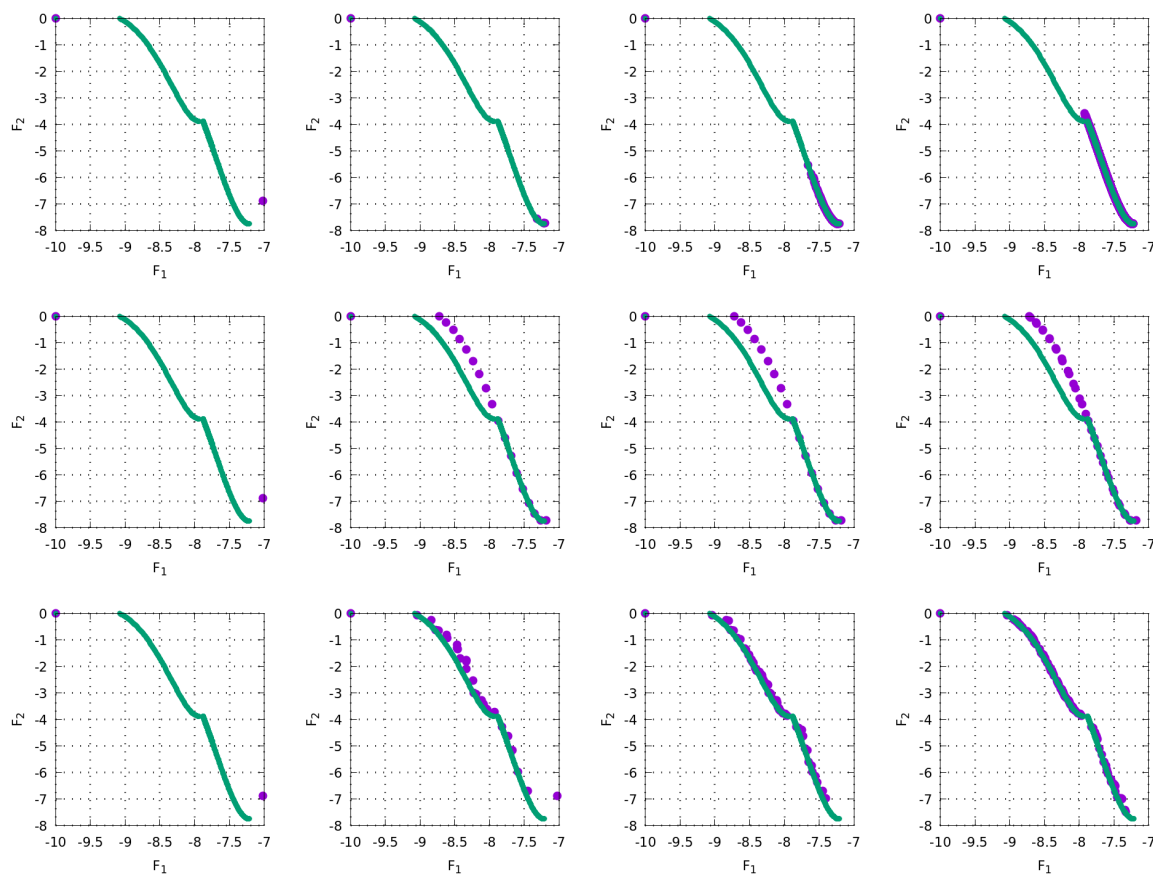
As a first test, a commonly used set of test functions has been adopted, namely the Kursawe functions. Here we have two objectives:

$$f_1(x) = \sum_{i=1}^{n-1} [-10 \exp(0.2 \sqrt{x_i^2 + x_{i+1}^2})]$$

$$f_2(x) = \sum_{i=1}^n [|x_i|^{0.8} + 5 \sin(x_i^3)]$$

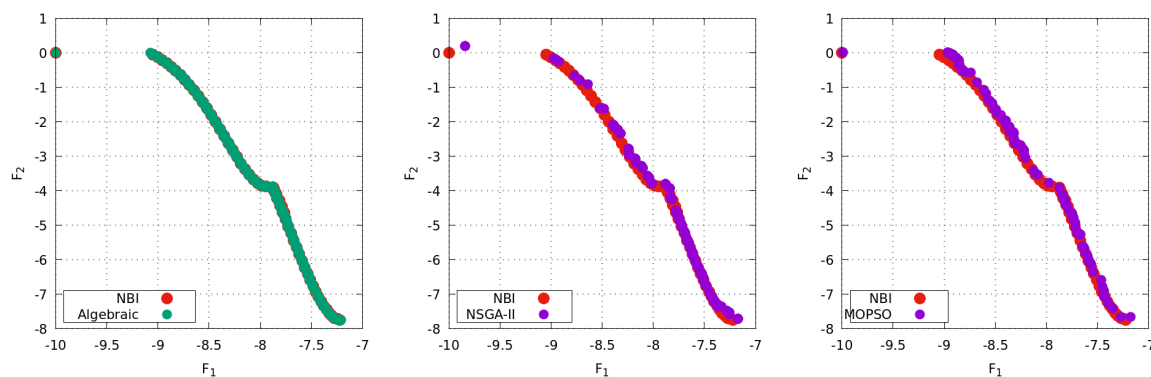
The number of design variables  $n$  is 2, and both the design variables can assume any value  $\in [-5 : 5]$ . No further constraints are applied.

A first test has been performed by applying the three methodologies separately. Results are reported in Figure 1: from top to bottom, we can observe the results obtained by applying only local resampling (top), only Pareto front interpolation (middle) and only NBI (bottom). Here is evident that Pareto front interpolation cannot be applied without another enrichment method: the Pareto front obtained at the first iteration is quite well developed, but it is obviously no further improved since the added points are always on the same line. Local resampling gives a variety of points and it is able to improve at each iteration, but it appears to be pretty slow in this case, and a branch of the curve is not obtained: this is probably connected with the quality of the initial sampling. NBI gives the more interesting results, with a continuous improvement, iteration by iteration.



**Figure 1.** Pareto front approximation obtained by using the three methodologies separately. From top to bottom, the results obtained by applying only local resampling (**top**), Pareto front interpolation (**middle**) and NBI (**bottom**). From left to right, four different stages of evolution of the methodology. Green dots are the true Pareto front.

Completely different results are obtained if the three methodologies are applied in sequence, with the same order as indicated previously: resampling, interpolation and lastly NBI. The effect of the combination of the three methodologies is reported in Figure 2. In the same picture, the results obtained by a standard MOGA implementation, namely NSGA-II [8], and a Multi-Objective Particle Swarm Optimization (MOPSO) algorithm [9] are also presented in order to produce a comparison with other standard methodologies. The same total number of objective function evaluations is fixed for all the algorithms, in order to produce a fair comparison. The first graph on the left of Figure 2 reports the comparison of the results obtained by NBI and the true Pareto front. Real Pareto front is produced by using a recursive intensive regular sampling of the design space. We have a perfect agreement of NBI with the true Pareto front, with an impressive uniformity of the distribution of points along the front. The other two pictures in Figure 2 report a comparison of NBI with other two standard multi-objective algorithms: NSGA-II in the center plot and MOPSO in the plot at the right end side of the picture. Both NSGA-II and MOPSO are producing good approximation of the Pareto front, but with a lack of precision with respect to NBI. NSGA-II is missing the best point for the first objective function, and in general both the methods do not present the same uniformity the NBI is able to provide. We can conclude that the NBI algorithm is much more efficient than the two selected algorithms, at least for this example.



**Figure 2.** Pareto front approximation obtained by using the three methodologies sequentially, in comparison with a different optimization algorithm (NSGA-II and MOPSO). In the first picture at the left, NBI is compared with the true Pareto front. The other two pictures show the comparison of the results obtained by NBI versus NSGA-II and MOPSO respectively.

#### 4. Results and Discussion

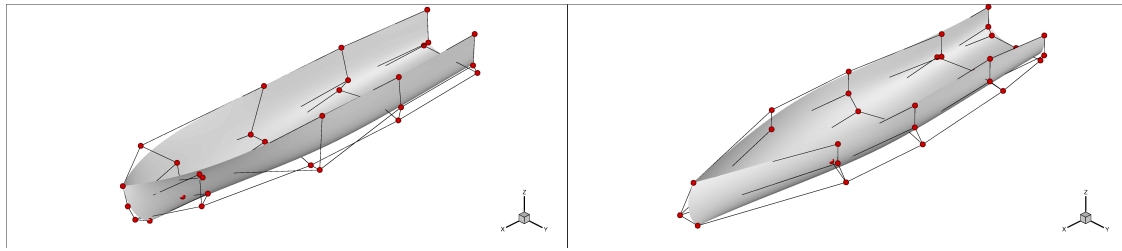
Two ship design applications are presented in the following, in order to give further elements demonstrating the efficiency of the approach. In the first example, a powering reduction problem for two different speeds is formulated and solved: a relatively simple but reliable Computational Fluid Dynamics (CFD) tool, based on the potential flow theory [10], is utilized. In the second example, the problem of the improving the quality of the flow at the propeller disk of a single-skew ship is solved by using a more sophisticated CFD solver. Different parameterization techniques are adopted in order to present different alternatives for the hull geometry modification.

##### 4.1. Ship Design Application: Powering

In order to further demonstrate the efficiency of the methodology, a second test case has been designed. For this test, the effective powers required by a ship advancing at two different speeds are the objectives of the multi-objective optimization problem. The two selected speeds are characterized by a nondimensional quantity (Froude number— $Fr$ ): this is to emphasize the different flow regimes at the two different speeds. With  $Fr = 0.3$  we are in the displacement regime, where the lifting actions of the flow on the ship are negligible. With  $Fr = 0.45$ , we are in the semi-displacement regime, and the lifting actions are more intense, of the same order of magnitude of the buoyancy actions. Although the adopted flow solver is modeling the lifting forces in a simplified way, the results provided by the solver are reliable in both the two regimes. Furthermore, the use of two different flow regimes is reflected on the optimal shape, different for the range of application: typically, semi-displacement ships have finer bow entry and flatter aft with respect to displacement ships. For this reason, we are expecting different optimal shapes for the two objectives, and than a large Pareto front.

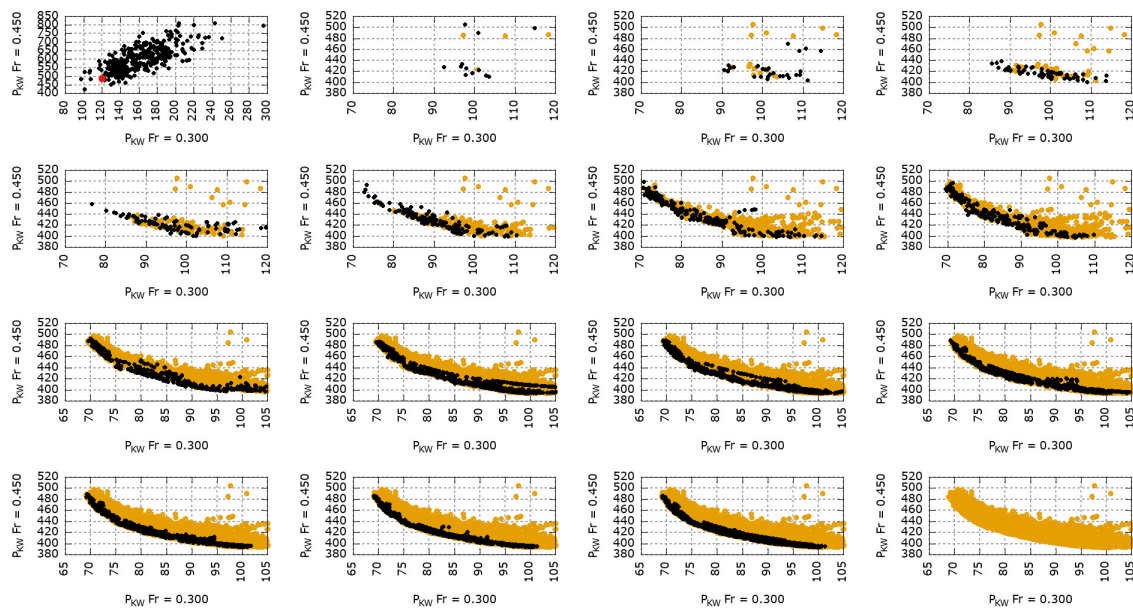
The parent hull to be modified and optimized is taken from the NPL series [11]. The 5A hull has been scaled up to an overall length 36 m, maximum width 6 m, moulded depth 4 m, mean draft when fully loaded 1.75 m, tonnage 200 tonnes. The hull surface has been replicated using a single Non-Uniform Rational Basis-Spline (NURBS—see the left image in Figure 3), and the control points represent the design parameters of the optimization problem. Every control point can move in every direction: the only fixed nodes are the ones at the bridge (locked in the vertical direction), at the stern (locked in the longitudinal direction) and the fore point on the bridge (locked in every direction). We have a total Number of Design Variables (NDV) of 21.



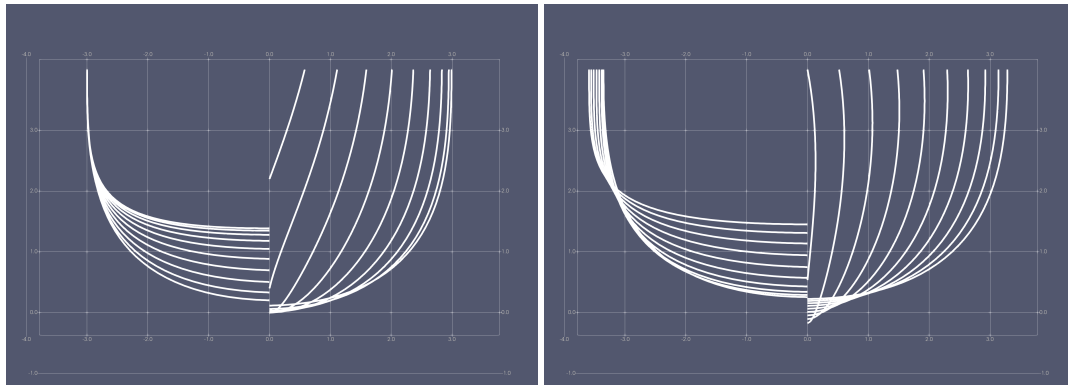


**Figure 3.** Representation of the hull surface adopted in the effective power optimization problem by means of a single NURBS: original shape (on left) and optimal shape (on right) obtained with a shift of the control points of the NURBS.

40 iterations of the full NBI procedure have been performed, evaluating a total number of 20,000 different configurations (around  $1000 \times NDV$ ): this number can be assumed as representative of a medium effort for a single-objective optimization problem. Results are reported in Figure 4. The top left picture of Figure 4 reports the initial approximation of the Pareto front after the initial sampling phase: it is obtained by using  $16 \times NDV$  solutions. Since then, in every picture the black dots are indicating the newly added solutions at every iteration. The central solution of the Pareto front is reported in the right part of Figure 3, and a comparison of the section views, original versus optimal, is reported in Figure 5. From this last picture we can observe the finer bow entry and flatter aft, as expected. We can see also in Figure 4 how the method is concentrating the newly added points around the current approximation of the true Pareto front: it seems that all the new solutions are not dispersed in a useless area, but they are all concentrated around the real Pareto front, improving its resolution at each iteration. This is an indication of the efficiency of the method.



**Figure 4.** Evolution of the Pareto front during the iterations of the multi-objective optimization procedure of the effective power. Black dots are the newly added points after one full iteration (all the three methodologies).



**Figure 5.** Comparison of the section views for the original (**left**) and the optimal (**right**) hull shapes for the effective power optimization problem.

#### 4.2. Ship Design Application: Propulsion

A second realistic ship design optimization problem has been solved using the proposed methodology. In this case, the flow at the propeller disk for a bulk carrier has been optimized. The parent hull form is represented by the Japan Bulk Carrier (JBC), whose geometry has been adopted as test case for several workshops<sup>3</sup>. Full loaded condition as reported in the workshop tests has been considered.

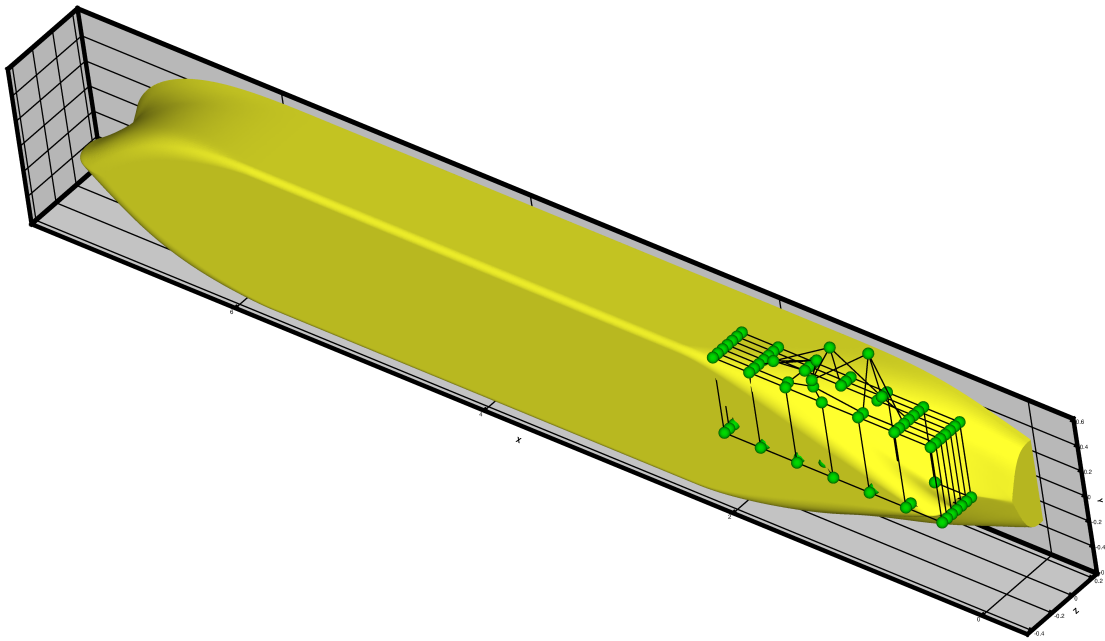
In order to obtain a more regular flow at the propeller disk, possibly improving the working conditions of the propeller, two quantities have been selected to be minimized: the alignment of the local flow with the advancing direction and the variance of the local speed vector module. The first objective function is measured by the average value of the scalar vector between the local speed and the advancing direction: since they are pointing in opposite directions, the value we hope to get is  $-1$ , so that also this objective is to be minimized.

Local flow is computed by using the suite OpenFoam v7: the solver is based on the the unsteady Reynolds-Averaged Navier-Stokes Equations (RANSE) model, implemented in the solver (interFoam) with k-Omega<sub>SST</sub> turbulence model. Simulation is stopped at 30 s of physical time: we have observed that solution becomes stable after around 25 s.

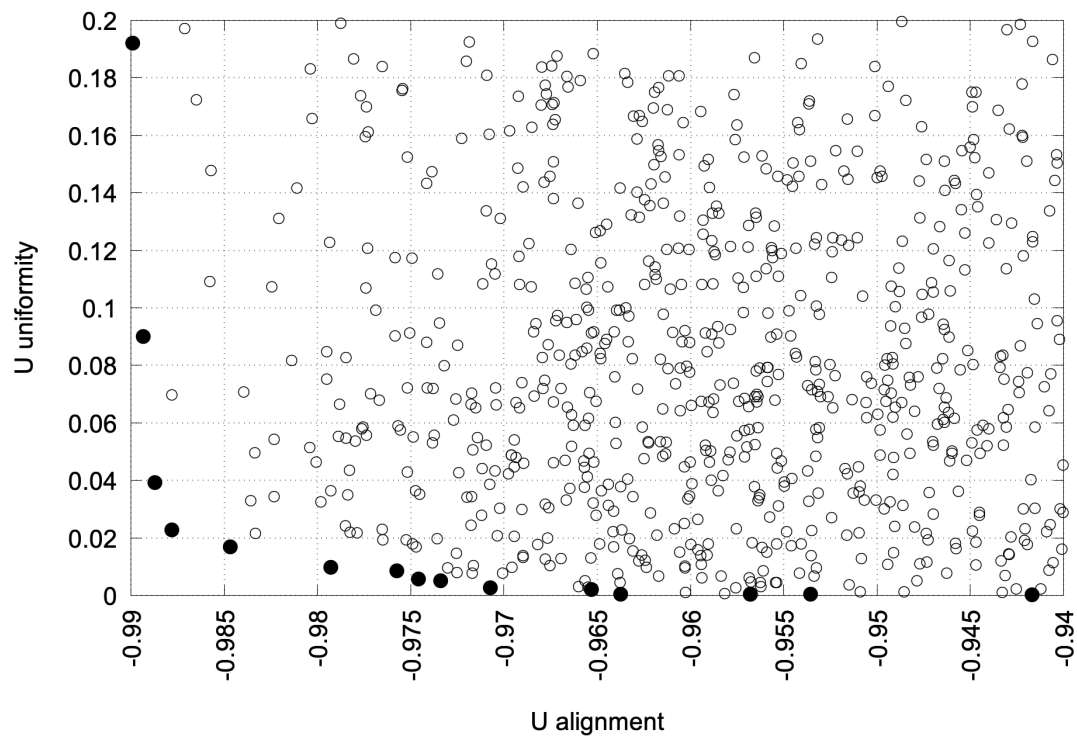
The modification of the hull has been restricted to the bossing area and the surrounding part. A portion of the aftbody has been included in order to facilitate the fairing of the resulting surfaces. The modification methodology is here the Free Form Deformation (FFD) [12,13]. An FFD with  $7 \times 2 \times 7$  subdivisions along the X, Y and Z axis respectively, has been placed in the area in front the propeller. In order to preserve the fairing of the hull surface, some of the nodes of the FFD are locked at the original position: the total number of design variables is 9, since only lateral movements are allowed in this case. A perspective view of the hull and the FFD box is reported in Figure 6. The method have produced 1350 alternatives ( $150 \times NDV$ ), a smaller number with respect to the previous example, since the computational effort is considerably higher. The resulting Pareto front is reported in Figure 7. Also in this case, the Pareto front (indicated with black dots) is rich and well distributed. For a comparison of the alternatives from the geometrical point of view, three shapes have been selected: the best solutions for each objective and an intermediate one, representing the best trade-off between the objectives. This last shape has been selected normalizing the objectives in between 0 and 1, with a proportional scaling, and than selecting from the Pareto front the solution closer to the bisector of the objective function space. The selected geometries are reported in Figure 8.

<sup>3</sup> for details, see [https://t2015.nmri.go.jp/jbc\\_gc.html](https://t2015.nmri.go.jp/jbc_gc.html).

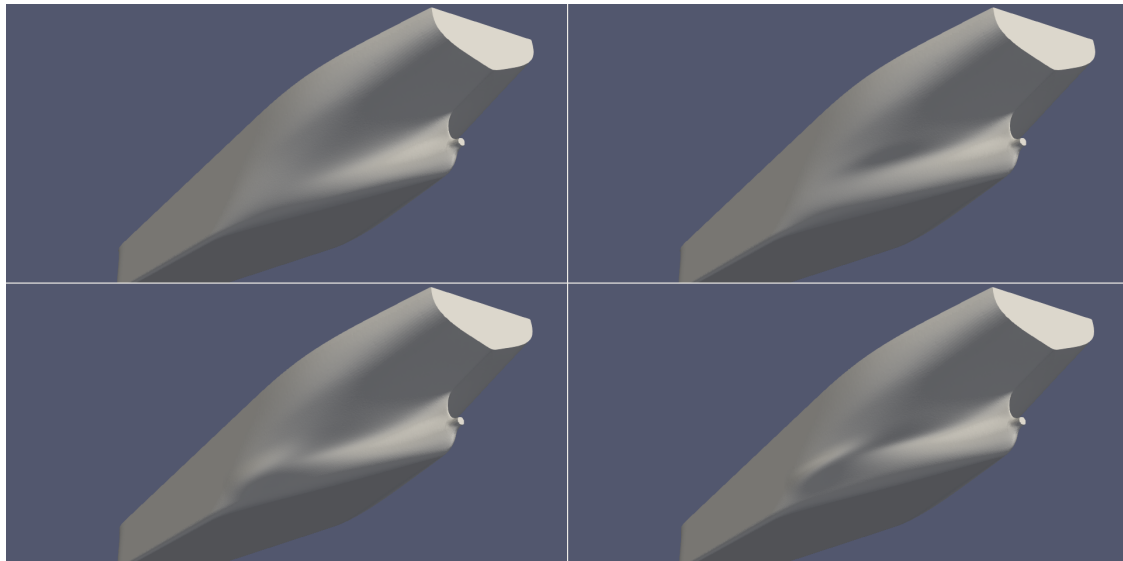




**Figure 6.** Perspective view of the hull surface (JBC) together with the FFD box for this hull parameterization. Green dots are representing the nodes of the FFD, and only 9 of them can move transversally.



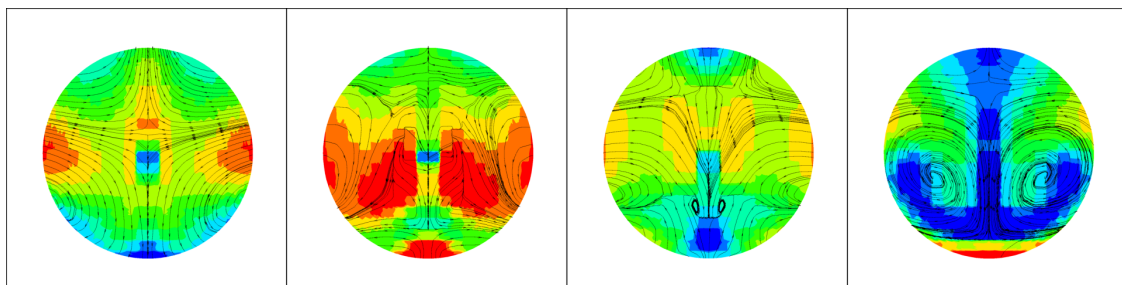
**Figure 7.** Pareto front for the multi-objective optimization problem of the JBC containership. Black dots are representing the computed Pareto front, white dots are indicating other tentative solutions.



**Figure 8.** Comparison of the original hull shape (**top left**) and three Pareto optimal solutions: the best for the first objective function (**top right**), a solution with the best balance between the two objectives (**bottom left**) and the best for the second objective (**bottom right**).

We can observe how the modification of the three alternative hulls is very similar, being the amount of the modification the only difference. The optimizer finds that the most promising area producing an appreciable variation of the flow at the propeller disk is the region of the aftbody just in front of the restriction of the hull generating the bossing area. A bump is generated in this part of the hull, and the height of the bump (and its sharpness) represents the main difference between the three hulls.

In Figure 9 we can observe the flow at the propeller disk for the original configuration and for the three alternatives. The corresponding values of the objectives are reported in Table 1. Here we can observe how clearly the geometry with the best value of the local speed alignment is also increasing the average value of the local flow (looking at the longitudinal component), while the intermediate solution (the second from right in Figure 9) has a quite uniform value of the longitudinal component of the local speed. A different situation is arising for the third shape, where the decrease of the variance is obtained at the expense of the average value, strongly reduced. Furthermore, we can also observe the occurrence of a negative side effect, that is, a strong vortex in the lower part of the propeller disk. From the methodological standpoint this is not representing a problem, since the Pareto front has been correctly identified, but practically this is representing a failure. The problem should be probably reformulated, including the average value of the local speed into the definition of the second objective function.



**Figure 9.** Flow at the propeller disk for four different configurations. From left to right, the original shape, the best shape for the first objective function, the configuration with the best balance between the two objectives and the best configuration for the second objective. Colors are representing the axial velocity of the flow, while the transverse component is indirectly reported by the streamtraces.

**Table 1.** Objective function values for some selected solutions.

|                 | $\bar{u}$ | $\sigma_u$ | $\bar{D}$ | $\sigma_D$ |
|-----------------|-----------|------------|-----------|------------|
| Original        | −0.737169 | 0.126924   | −0.951803 | 0.001652   |
| Best $\bar{D}$  | −0.822615 | 0.192103   | −0.989913 | 0.000016   |
| Intermediate    | −0.734401 | 0.001688   | −0.984661 | 0.000876   |
| Best $\sigma_u$ | −0.603269 | 0.000019   | −0.941700 | 0.000520   |

## 5. Conclusions

A new methodology for the determination of the Pareto front in a multi-objective optimization problem has been formulated. Although the three components of the full methodology are not able to provide good results if applied in isolation, the combined use has been demonstrated to be very efficient. Application to algebraic functions and realistic applications gives the measure of the efficiency of the methodology: a dense and regular Pareto front is provided with an expense similar to the one of the solution of a single-objective optimization problem. The comparison with a standard MOGA highlights the good performances of the algorithm, being the results of the combined NBI procedure outperforming the MOGA. The solution of the two ship design applications is also giving clear indications about the ability of the proposed methodology to correctly identify the full Pareto front at the expense of a moderate computational effort, also in this case similar to the one required by a standard single objective optimization process.

One of the selected applications show some shortcomings in the definition of one of the objective functions, and it could be interesting to repeat the computations in order to observe the effects on the final configuration. Also the comparison with a single different methodology is not enough, and a more extensive and complete comparison with more similar algorithms has to come.

New design applications could also be formulated and solved in order to generalize the actual results, also increasing the number of objectives.

**Funding:** This research was funded by Italian Minister of Instruction, University and Research (MIUR) to support this research with funds coming from PRIN Project 2017 (No. 2017KKJP4X entitled “Innovative numerical methods for evolutionary partial differential equations and applications”).

**Conflicts of Interest:** The author declare no conflict of interest.

## References

1. Cui, Y.; Geng, Z.Q.; Zhu, Q.X.; Han, Y.M. Review: Multi-objective optimization methods and application in energy saving. *Energy* **2017**, *125*, 681–704. [\[CrossRef\]](#)
2. Das, I.; Dennis, J.E. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Struct. Optim.* **1997**, *14*, 63–69. [\[CrossRef\]](#)
3. Goldberg, D.E. Genetic Algorithms in Search. In *Optimization and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989.
4. Sobol, I.M. Distribution of points in a cube and approximate evaluation of integrals *Zh. Vych. Mat. Mat. Fiz.* **1967**, *7*, 784–802. (In Russian)
5. Beachkofski, B.; Grandhi, R. *Improved Distributed Hypercube Sampling*; Paper 1274; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2002.
6. Vandenberghe, L.; Boyd, S.; Wu, S.P. Determinant maximization with linear matrix inequality constraints. *SIAM J. Matrix Anal. Appl.* **1998**, *19*, 499–533. [\[CrossRef\]](#)
7. Peri, D. Easy-to-implement multidimensional spline interpolation with application to ship design optimisation. *Ship Technol. Res.* **2018**, *65*, 32–46. [\[CrossRef\]](#)
8. Kasat, R.B.; Gupta, S.K. Multi-Objective Optimisation of an Industrial Fluidized-Bed Catalytic Cracking Unit (FCCU) Using Genetic Algorithm (GA) with the Jumping Genes Operator. *Comput. Chem. Eng.* **2003**, *27*, 1785–1800. [\[CrossRef\]](#)
9. Pinto, A.; Peri, D.; Campana, E.F. Multiobjective Optimization of a Containership Using Deterministic Particle Swarm Optimization. *J. Ship Res.* **2007**, *51*, 217–228.

10. Gadd George, E. A method for calculating the flow over ship hulls. *Trans. Inst. Nav. Archit.* **1970**, *112*, 335–345.
11. Bailey, D. The NPL High Speed Round Bilge. In *Displacement Hull Series: Resistance, Propulsion, Manoeuvring and Seakeeping Data*; Royal Institution of Naval Architects: London, UK, 1967.
12. Sederberg, T.W.; Parry, S.R. Free-form deformation of solid geometric models. In Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, Dallas, TX, USA, 18–22 August 1986; Volume 20, pp 151–160.
13. Tahara, Y.; Peri, D.; Campana, E.F.; Stern, F. Computational fluid dynamics-based multi-objective optimization of a surface combatant using a global optimization method. *J. Mar. Sci. Technol.* **2008**, *13*, 95–116. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).