Università degli studi “Roma Tre”

Department of Mechanical and Industrial Engineering

Cycle XXVIII

Hybrid global/local optimization methods in simulation-based shape design

PhD candidate
Andrea Serani

Tutor
Prof. Umberto Iemma

CNR-INSEAN Tutor
Dr. Matteo Diez

June 2016

To Valentina

Acknowledgements

First and foremost I want to thank my CNR-INSEAN tutor Dr. Matteo Diez. It has been an honour to be his first Ph.D. student. I appreciate all his contributions of time and ideas shared to make my Ph.D. experience productive and stimulating. The joy and enthusiasm he has for his research is contagious and motivational for me. I am also thankful for the excellent example he has provided as a successful researcher.

A special thanks goes to my colleague Dr. Cecilia Leotardi for her support provided during the last three years. I am grateful to my university tutor Prof. Umberto Iemma, to the CNR-INSEAN director Dr. Emilio F. Campana, for his support through the CNR-INSEAN fellowship, and to Prof. Frederick Stern of the IIHR-Hydroscience & Engineering laboratory for his supervision during the three months that I spent as visiting Ph.D. student at the University of Iowa.

This work has been partially performed in collaboration with NATO STO Task Groups AVT-204 "Assess the Ability to Optimize Hull Forms of Sea Vehicles for Best Performance in a Sea Environment" and AVT-252 "Stochastic Design Optimization for Naval and Aero Military Vehicles.". I am grateful to Dr. Woei-Min Lin and Dr. Ki-Han Kim of the US Navy Office of Naval Research, for their support through NICOP grant N62909-15-1-2016 and grant N00014-14-1-0195. I am also grateful to the Italian Flagship Project RITMARE, coordinated by the Italian National Research Council and founded by the Italian Ministry of Education.

Abstract

Simulation-based design optimization methods integrate computer simulations, design modification tools, and optimization algorithms. In hydrodynamic applications, often objective functions are computationally expensive and noisy, their derivatives are not directly provided, and the existence of local minima cannot be excluded a priori, which motivates the use of deterministic derivative-free global optimization algorithms. DPSO (Deterministic Particle Swarm Optimization), DIRECT (DIviding RECTangles), two well-known derivative-free global optimization algorithms, and FSA (Fish Shoal Algorithm), a novel metaheuristic introduced herein, are described in the present work. Moreover, the enhancement of DIRECT and DPSO is presented based on global/local hybridization with derivative-free line search methods. DPSO, DIRECT, FSA, and three hybrid algorithms (LS-DF_PSO, DIRMIN, and DIRMIN-2) are introduced, and assessed on a benchmark of seventy-three analytical functions, providing an effective and efficient guideline for their use in the simulation-based shape design optimization context. The suggested guidelines are applied on ten hull-form optimization problems, using potential flow and RANS solvers, supported by metamodels. The optimizations pertain the high-speed Delft catamaran and an USS Arleigh Burke-class destroyer ship, namely the DTMB 5415 model, an early and open-to-public version of the DDG-51. Three shape modification techniques, specifically the free-form deformation and the orthogonal basis functions expansion over 2D and 3D subdomains, are introduced, along with the design-space dimensionality reduction by generalized Karhunen-Loève expansion. Hybrid algorithms show a faster convergence towards the global minimum than the original global methods and therefore represent a viable option for shape design optimization. Moreover, FSA shows a better effectiveness compared to the other global algorithm (DPSO and DIRECT), allowing for good expectations for its further improvement with a local hybridization, for the future work.

Contents

Nomenclature

$n \in \mathbb{N}^+$	algorithm iteration counter
$N \in \mathbb{N}^+$	number of (design) variable
$N_{\max} \in \mathbb{N}^+$	maximum number of function evaluations
$N_p \in \mathbb{N}^+$	number of particles (for PSO-based algorithm)
$N_s \in \mathbb{N}^+$	number of individuals (for FSA algorithm)
$\mathbf{x} \in \mathbb{R}^N$	design variable vector
$\mathbf{l} \in \mathbb{R}^N$	$\mathbf{x}$ lower bound
$\mathbf{u} \in \mathbb{R}^N$	$\mathbf{x}$ upper bound
$f(\mathbf{x}) \in \mathbb{R}$	objective function
$h(\mathbf{x}) \in \mathbb{R}$	equality constraint
$g(\mathbf{x}) \in \mathbb{R}$	inequality constraint
$\boldsymbol{\delta}_s \in \mathbb{R}$	shape design modification vector
$\nabla \in \mathbb{R}$	displacement
$B \in \mathbb{R}$	beam overall
$\mathrm{Fr} \in \mathbb{R}$	Froude number
$H_w \in \mathbb{R}$	wave height
$K_{xx} \in \mathbb{R}$	roll radius of gyration
$K_{yy} \in \mathbb{R}$	pitch radius of gyration
$K_{zz} \in \mathbb{R}$	yaw radius of gyration
$LCG \in \mathbb{R}$	longitudinal center of gravity
$LBP \in \mathbb{R}$	length between perpendicular
$LOA \in \mathbb{R}$	length overall
$\mathrm{Re} \in \mathbb{R}$	Reynolds number
$T \in \mathbb{R}$	draft
$T_w \in \mathbb{R}$	wave period
$VCG \in \mathbb{R}$	vertical center of gravity
A-DPSO	asynchronous deterministic particle swarm optimization
CFD	computational fluid dynamics
DM	Dawson (double model) linearization
DC	Delft catamaran
DIRECT	dividing rectangle
DOF	degrees of freedom
DPSO	deterministic particle swarm optimization
DTMB	David-Taylor model basin
EFD	experimental fluid dynamics
EW	elastic wall
FEM	finite element method
FFD	free-form deformation
FSA	fish shoal algorithm
HSS	Hammersley sequence sampling
IW	inelastic wall
KLE	Karhunen-Loève expansion
LS-DF_PSO	line-search derivative-free particle swarm optimization
NK	Neumann-Kelvin linearization

OBFE	orthogonal basis functions expansion
PF	potential flow
PSO	particle swarm optimization
PSS	positively spanning set
RANS	Reynolds-averaged Navier-Stokes
S-DPSO	synchronous deterministic particle swarm optimization
SBDO	simulation-based design optimization
SEW	semi-elastic wall
URANS	unsteady Reynolds-averaged Navier-Stokes

Introduction

"No, no! The adventures first, explanations take such a dreadful time."

— Lewis Carroll, *Through the Looking-Glass*

In the last decades, engineering design practices have experienced a paradigm shift, moving from the traditional build-and-test approach to more efficient, effective and versatile simulation-based design (SBD) methodologies. The integration of optimization algorithms with computer simulations has led to simulation-based design optimization (SBDO) procedures, with the aim of assisting and, if possible, guiding the designer in the the decision making process of complex engineering applications. The complexity of the systems along with the need of accurate performance analyses have led to the development and application of high-fidelity simulation codes, based on prime principles. Systems of partial differential equations are generally solved by computationally expensive black-box tools, such as those used in computational fluid dynamics (CFD) or structural finite element methods (FEM). The ever-increasing demand for accuracy and the complexity of structures and systems results to be more and more time consuming, in the simulation process. In several engineering fields, the evaluation of a single design can take as long as many days or even weeks. For this reason, new methods that can speed up the simulation time and the optimization process, saving time and money, are sought-after. SBD optimization procedures have been developed in order to integrate, efficiently and effectively, numerical simulations, design modification tools, and optimization algorithms (see Fig. 1). For this reason, SBDO is actually an essential part of the design of complex engineering systems since the conceptual and early design stages. Despite dramatic advances in computer hardware and software performance, the goal of fast, flexible and accurate simulation and optimization is yet to be achieved. Creating a system that has acceptable performance and provides useful results is a significant challenge.

There are plenty of challenging real applications in sciences where optimization is naturally involved, and sophisticated minimization techniques are definitely necessary in order to allocate resources. SBD optimization for shape design has been widely applied to several engineering fields, such as aerospace [1, 2, 3, 4, 5], automotive [6, 7, 8, 9], structural [10, 11] and naval [12, 13, 14, 15, 16] engineering, where the shape design is of primary importance for the vehicle performance (e.g., aerodynamic, aeroelasticity, flight mechanics, hydrodynamics, seakeeping, structures, heat transfer). SBDO methodologies generally require large computational simulations to assess the performance of a design and evaluate the relative merit of design alternatives. In this context, an automated SBDO needs to integrate (i) simulation tools (for structures, fluids, etc.) and (ii) minimization algorithms with (iii) geometry modification and automatic meshing algorithms. To obtain an automated process this three fundamental elements have to be linked together in a robust, efficient, and effective way (see Fig. 1).

Despite their importance, there are no satisfactory rules or guidelines for such issues. Obviously, the actual efficiency of an algorithm depends on many factors such as the inner working of an algorithm, the information needed (such as objective functions and their derivatives), and implementation details. The efficiency of a solver is even more complicated, depending on the actual numerical methods used and the complexity of the problem of interest. As for choosing the right algorithms for the right problems, there are many empirical observations, but no agreed guidelines. In fact, there is no universally efficient algorithms for all types of problems [17]. Therefore, the choice depends on several factors and is sometimes subject to the personal preferences of researchers and decision makers.

Within SBDO, a nonconvex nonlinear programming problem is solved, where the objective function represents the performance of the engineering system under analysis and is usually of the black-box type, with values provided by computationally-expensive computer simulations. Up to 15-20 years ago, to a large extent, the main interest of theoreticians in optimization was for methods based on the use of derivatives. This was basically due to the following three strong reasons:

- in several cases derivatives are available when solving computational problems. In particular, they are always 'analytically' available if the nonlinear functions involved are known in closed form [18]), and they can be exactly computed (not simply approximated) at reasonable cost in small-medium scale problems [19, 20];

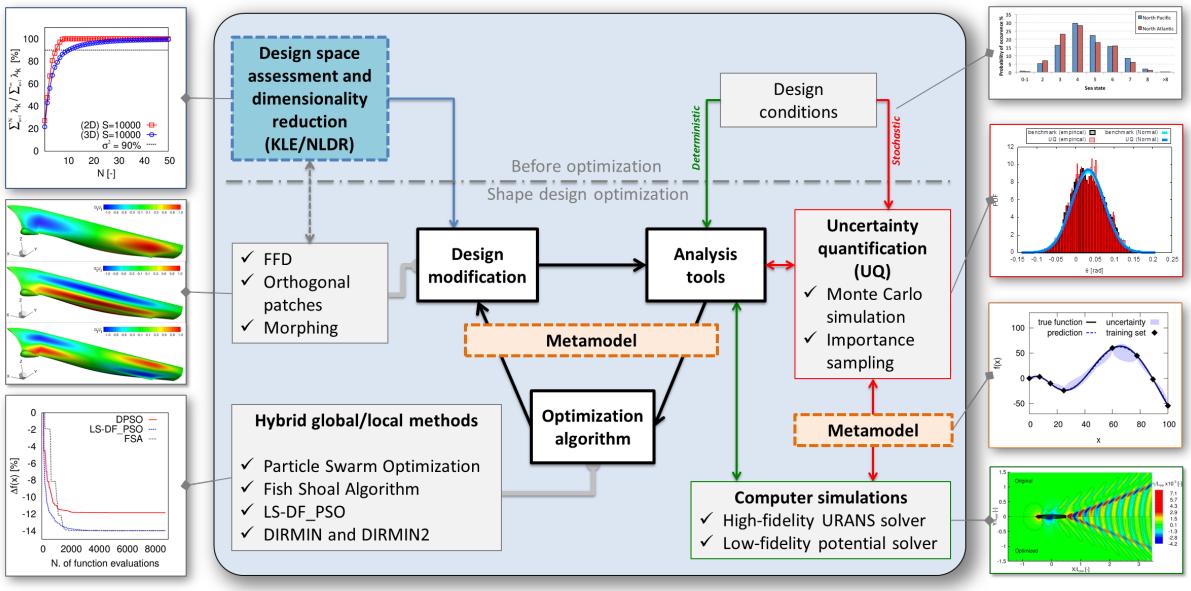


Figure 1: Simulation-based design optimization framework

- strong theoretical results have been developed, both in terms of convergence and computational performance, for optimization methods where derivatives (say of first/second order) are available;

- the use of machine resources at a cheaper cost has allowed the solution of problems where derivatives can be suitably approximated by finite differences, using either coarse or fine techniques.

On the other hand, engineering design offers a huge number of real-world problems where scientists are continuously asked to apply robust methods, using the most recent theoretical advances. In particular, design problems often include functions which are non-differentiable or where the use of derivatives is possibly discouraged. The following issues motivate the latter statement and give more precise guidelines for analyzing and improving optimization procedures not involving derivatives.

- For 'large scale' problems, computing derivatives by finite differences might be prohibitively costly, and also Automatic Differentiation [18] might be of difficult application. Furthermore, the computation of derivatives by finite differences proved to be very harmful when the scale of the problem increases. Potential design improvements significantly depend on dimension and extension of the design space: high dimension and variability spaces are more difficult and expensive to explore but, at the same time, potentially allow for bigger improvements.

- Most of the codes for complex design problems are 'parameter dependent', and the parameters need to be properly assessed. Their correct choice in practice implies that the overall performance of the code needs to be optimized with respect to those parameters. Thus, an implicit optimization problem with respect to these parameters requires a solution, and surely the derivatives of the functions involved are unavailable, being the output of a non-differentiable code.

- Most of the design problems need solution procedures where expensive simulations are performed. Typically, simulations are affected by 'noise', systematic errors arise and stochastic parameters are used, so that derivatives are essentially unavailable or their use may lead to completely destroy the robustness of procedures [21].

The aforementioned issues contribute to motivate the use of efficient and effective derivative-free global methods, in order to solve a wide range of challenging problems. Derivative-free global optimization algorithms have been developed and effectively applied to SBD optimization, providing global approximate solutions to the design problem [22, 23, 24]. Moreover, shape optimization research has focused on shape and topology parameterizations, as a critical issue to achieve the desired level of design variability, and some recent research focused on research space variability and dimensionality reduction for efficient analysis and optimization procedures [5, 25].

When global techniques are used with CPU-time expensive solvers, the optimization process is computationally expensive and its effectiveness and efficiency remain an algorithmic and technological challenge. Although complex SBD applications are often solved by metamodels [26, 27], their development and assessment require benchmark solutions, with simulations directly connected to the optimization algorithm. These solutions are achieved only if affordable and effective optimization algorithms are available. Several global optimization algorithm, such as simulated annealing (SA) [28, 29], particle swarm optimization (PSO) [30], ant colony optimization (ACO) [31], and genetic algorithms (GA) [32] have been deeply investigated in the last years [33, 34], and several new algorithms appeared recently, such as artificial fish-swarm algorithm (AFSA) [35], mesh adaptive direct search (MADS) [36, 37], firefly algorithm (FA) [38], cuckoo search (CS) [39], and bat algorithm (BA) [40]. Derivative-free global optimization approaches are often preferred to local approaches when objectives are non-convex and/or noisy, and when multiple local optima cannot be excluded, as often encountered in SBDO. Although global optimization approaches are a good compromise between exploration and exploitation of the research space, they could still get trapped in local minima and the convergence to a global minimum cannot be proven. If the research region to explore is known a priori, local optimization approaches can give an accurate approximation of the local minimum. Nevertheless, their convergence may be computationally expensive, and the information is usually not available a priori. For these reasons, the hybridization of global optimization algorithms with local search methods is an interesting research field, especially if CPU-time expensive black-box functions are involved, where the qualities of both methods can be efficiently and robustly coupled. It is worth noting that a large variety of derivative-free global and local methods available in the literature are stochastic/probabilistic. These methods make use of random coefficients and have been developed to the aim of sustaining the variety of the search for an optimum. This property implies that statistically significant results can be obtained only through extensive numerical campaigns. Such an approach can be too expensive (often almost unaffordable) in SBD optimization for industrial applications, when CPU-time expensive computer simulations are used directly as analysis tools. For this reason, deterministic approaches have been successfully developed and applied to SBD optimization.

The objective of the present work is introduce, assess, and apply several optimization and geometry modification techniques for an efficient and effective use in the SBDO context. Chapter 1 introduce six derivative-free global and global/local optimization algorithms, suitable for an efficient and effective application in the SBDO context. Chapter 2 defines the three shape deformation methodologies used herein and the dimensionality reduction concept. The numerical solvers used for naval engineering application are presented in Chapter. 3. Chapter 4 presents the optimization algorithm parameters to be assess on seventy-three analytical test function, in order to define a useful guideline for each algorithm, and the naval engineering SBDO benchmark problem, used to verify and compare the algorithms performance. The results obtained on the test functions and the naval SBDO problems are presented and discussed in Chapter 5. Finally, Chapter 6 is devoted to the conclusions remark and the possible future work related to this thesis.

Chapter 1

Optimization algorithms

"Begin at the beginning" the King said gravely,
"and go on till you come to the end: then stop"

— Lewis Carroll, *Alice in Wonderland*

Optimization problems can be formulated in several ways. The best-known formulation is to write a nonlinear optimization problem as

$$\begin{array}{rll} \text{Minimize} & f_i(\mathbf{x}), & i = 1,\ldots,M \\ \text{subject to} & h_j(\mathbf{x}) = 0, & j = 1,\ldots,J \\ \text{and to} & g_k(\mathbf{x}) \leq 0, & k = 1,\ldots,K \end{array} \tag{1.1}$$

where f_i, h_j, and g_k are general nonlinear functions. Here, the design vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ can be continuous, discrete, or mixed in N-dimensional space. The functions f_i are the objective or cost functions, whereas h_j and g_k are equality and inequality constraints, respectively. When $M > 1$, the optimization is multi-objective or multi-criteria. It is possible to combine different objectives into a single objective, though multi-objective optimization can give far more information and insight into the problem. The present work deals only with single-objective optimization algorithm.

Equation 1.1 represents in general a Non-deterministic polynomial-time hard (NP-hard) problem and no efficient (in the polynomial sense) solutions exist for a given problem. Thus, the challenges of research in computational optimization and applications are to find the most suitable algorithms for a given problem in order to obtain good solutions (perhaps also the best solutions globally), in a reasonable timescale with a limited amount of resources.

An efficient optimizer is very important to ensure the optimal solutions are reachable. There are several optimization algorithms in the literature, and no single algorithm is suitable for all problems, as dictated by the No Free Lunch Theorems [17].

Optimization algorithms can be classified in many ways, depending on the characteristics that are compared. Algorithms can be classified as gradient-based (or derivative-based) and gradient-free (or derivative-free). The classic methods of steepest descent and the Gauss-Newton methods are gradient-based, as they use the derivative information in the algorithm, whereas the Nelder-Mead downhill simplex method [41] is a derivative-free method because it uses only the values of the objective, not any derivatives. Algorithms can also be classified as deterministic or stochastic. If an algorithm works in a mechanically deterministic manner without any random nature, it is called deterministic. For such an algorithm (i.e. downhill simplex methods), it will reach the same final solution if it starts with the same initial point. On the other hand, if there is some randomness in the algorithm, the algorithm will usually reach a different point every time it is run, even starting with the same initial point (i.e. genetic algorithms). This property implies that statistically significant results can be obtained only through extensive numerical campaigns.

From the mobility point of view, algorithms can be classified as local or global. Local search algorithms typically converge toward a local optimum, not necessarily the global optimum, and such algorithms are often deterministic and have no ability of escaping local optima. On the other hand, the optimization procedure always try to find the global optimum for a given problem, and if this global optimality is robust, it is often the best, though it is not always possible to find such global optimality [42]. For global optimization [43], local search algorithms are usually not suitable. Modern metaheuristic algorithms, in most cases, are intended for global optimization.

Although global optimization approaches are a good compromise between exploration and exploitation of the research space, they could still get trapped in local minima and the convergence to a global minimum cannot be

proven. If the research region to explore is known a priori, local optimization approaches can give an accurate approximation of the local minimum, nevertheless, their convergence may be computationally expensive, and the information is usually not available a priori. Global/local hybrid algorithms tries to couple efficiently and robustly the qualities of both this methods.

From the optimization point of view, the choice of the right optimizer or algorithm for a given problem is crucially important. The algorithm chosen for an optimization task will largely depend on the type of the problem, the nature of an algorithm, the desired quality of solutions, the available computing resource, the time limit, the availability of the algorithm implementation, and the expertise of the decision makers. The nature of an algorithm often determines if it is suitable for a particular type of problem. For example, gradient-based algorithms are not suitable for an optimization problem with a discontinuous objective. Generally, if the objective function of an optimization problem at hand is highly nonlinear and multimodal, gradient-based algorithm are inappropriate, whereas global optimizers are more suitable.

This chapter provide a brief description of six single-objective deterministic derivative-free global and hybrid global/local optimization algorithm suitable for SBD global optimization problem in ship hydrodynamics. Two algorithms are well-known global optimization approaches, specifically a deterministic version of the particle swarm optimization method (DPSO) [44] and the DIviding RECTangles (DIRECT) algorithm [45]. Other three algorithms are hybrid global/local techniques integrated in DPSO and DIRECT enhancing the global methods with proved stationarity of the final solution; a hybrid DPSO coupled with line search-based derivative-free optimization (LS-DF PSO) [23], two hybrid DIRECT method coupled with line search-based derivative-free optimization (DIRMIN and DIRMIN-2) [24]. The last one is a novel metaheuristic based on the dynamics a fish shoal in search for food, namely the fish shoal algorithm (FSA) [46].

1.1 Particle Swarm Optimization algorithm

Particle Swarm Optimization (PSO) was originally introduced in Ref. [30], based on the social-behavior metaphor of a flock of birds or a swarm of bees searching for food. PSO belongs to the class of heuristic algorithms for single-objective evolutionary derivative-free global optimization.

The original PSO makes use of random coefficients, aiming at sustaining the variety of the swarm's dynamics. This property implies that statistically significant results can be obtained only through extensive numerical campaigns. Such an approach can be too expensive in SBD optimization for industrial applications, when CPU-time expensive (high-fidelity) computer simulations are used directly as analysis tools. For these reasons efficient deterministic approaches (such as deterministic PSO, DPSO) have been developed, and their effectiveness and efficiency in industrial applications in ship hydrodynamics problems have been shown, including comparisons with local methods [22] and random PSO [47]. Moreover, the availability of parallel architectures and high performance computing (HPC) systems has offered the opportunity to extend the original *synchronous* implementation of DPSO (S-DPSO) to CPU-time efficient *asynchronous* methods (A-DPSO) [48, 49].

1.1.1 PSO formulation

The original formulation of the PSO algorithm, as presented in [50], reads

$$\begin{cases} \mathbf{v}_j^{n+1} = w\mathbf{v}_j^n + c_1 r_{1,j}^n (\mathbf{x}_{j,pb} - \mathbf{x}_j^n) + c_2 r_{2,j}^n (\mathbf{x}_{gb} - \mathbf{x}_j^n) \\ \mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \mathbf{v}_j^{n+1} \end{cases} \tag{1.2}$$

The above equations update velocity and position of the j-th particle at the n-th iteration, where w is the *inertia weight*; c_1 and c_2 are the social and cognitive learning rate; $r_{1,j}^n$ and $r_{2,j}^n$ are uniformly distributed random numbers in $[0,1]$; $\mathbf{x}_{j,pb}$ is the *personal best* position ever found by the j-th particle and $\mathbf{x}_{gb}$ is *global best* position ever found considering all particles.

An overall *constriction factor* χ is used in [51, 52, 53, 54, 55], instead of the inertia weight w. Accordingly, the system in Eq. 1.2 is recast in the following equivalent form

$$\begin{cases} \mathbf{v}_j^{n+1} = \chi \left[\mathbf{v}_j^n + c_1 r_{1,j}^n (\mathbf{x}_{j,pb} - \mathbf{x}_j^n) + c_2 r_{2,j}^n (\mathbf{x}_{gb} - \mathbf{x}_j^n) \right] \\ \mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \mathbf{v}_j^{n+1} \end{cases} \tag{1.3}$$

In order to provide necessary (but possibly not sufficient) conditions which avoid divergence of particles trajectories, the following condition:

$$\chi = \frac{2}{\left| \sqrt{2-\varphi-\sqrt{\varphi^2-4\varphi}} \right|}, \qquad \text{where} \quad \varphi = c_1 + c_2, \ \varphi > 4 \tag{1.4}$$

is indicated in [52], where setting the value of φ to 4.1, with $\chi = 0.729$, $c_1 = c_2 = 1.494$ is suggested. Note that PSO schemes including both the parameters w and χ have been also proposed in the literature.

1.1.2 DPSO formulation

In order to make PSO more efficient and repeatable for use within SBD, a deterministic version of the algorithm (DPSO) was formulated in [22] by setting $r^n_{1,j} = r^n_{2,j} = 1$ in Eq. 1.3, which become

$$\begin{cases} \mathbf{v}_j^{n+1} = \chi \left[\mathbf{v}_j^n + c_1(\mathbf{x}_{j,pb} - \mathbf{x}_j^n) + c_2(\mathbf{x}_{gb} - \mathbf{x}_j^n) \right] \\ \mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \mathbf{v}_j^{n+1} \end{cases} \tag{1.5}$$

In the context of SBD for ship design optimization, as mentioned before, the formulation of Eq. 1.5 was compared to the original in [47]. DPSO is therefore used for all the subsequent analyses.

Using the above formulation, it is possible to prove that the necessary (but possibly not sufficient) conditions which ensure that the trajectory of each particle does not diverge [56], is

$$\begin{cases} 0 < |\chi| < 1 \\ 0 < \omega < 2\,(\chi + 1) \end{cases} \tag{1.6}$$

where $\omega = \chi(c_1 + c_2)$. Introducing

$$\beta = \frac{\omega}{2(\chi + 1)} \tag{1.7}$$

and assuming $\chi > 0$ as usually in the literature, the conditions of Eq. 1.6 reduce to

$$\begin{cases} 0 < \chi < 1 \\ 0 < \beta < 1 \end{cases} \tag{1.8}$$

1.1.3 Synchronous and asynchronous implementations

The synchronous implementation of DPSO (S-DPSO) updates the personal bests $\{\mathbf{x}_{j,pb}\}$ and the global best $\mathbf{x}_{gb}$, along with particles velocity and position, at the end of each iteration. S-DPSO is presented as a pseudo-code in Alg. 1, and as a block diagram in Fig. 1.1a.

Algorithm 1 S-DPSO pseudo-code

```
Initialize a swarm of N_p particles
while (n < Max number of iterations) do
    for j = 1, N_p do
        Evaluate f(x_j^n)
    end for
    Update {x_{j,pb}}, x_gb
    Update particle positions and velocities {x_j^{n+1}}, {v_j^{n+1}}
end while
Output the best solution found
```

In parallel architectures, if the amount of time required to evaluate the objective function at each iteration is not uniform (e.g., due to iterative process/convergence of analysis tools), the wall-clock time and CPU-time reservation of S-DPSO may significantly increase. In contrast to S-DPSO, the asynchronous implementation A-DPSO updates personal and global bests, along with particles velocity and position, as soon as the information required for their update is available, and a particle is ready for a new analysis. A-DPSO is presented as a pseudo-code in Alg. 2, and as a block diagram in Fig. 1.1b, where it can be seen the different CPU-time required for each objective function evaluation.

Algorithm 2 A-DPSO pseudo-code

```
Initialize a swarm of N_p particles
while (n < Max number of iterations) do
   for j = 1, N_p do
      Evaluate f(x_j^n)
      Update {x_{j,pb}}, x_{gb}
      Update particle positions and velocities {x_j^{n+1}}, {v_j^{n+1}}
   end for
end while
Output the best solution found
```

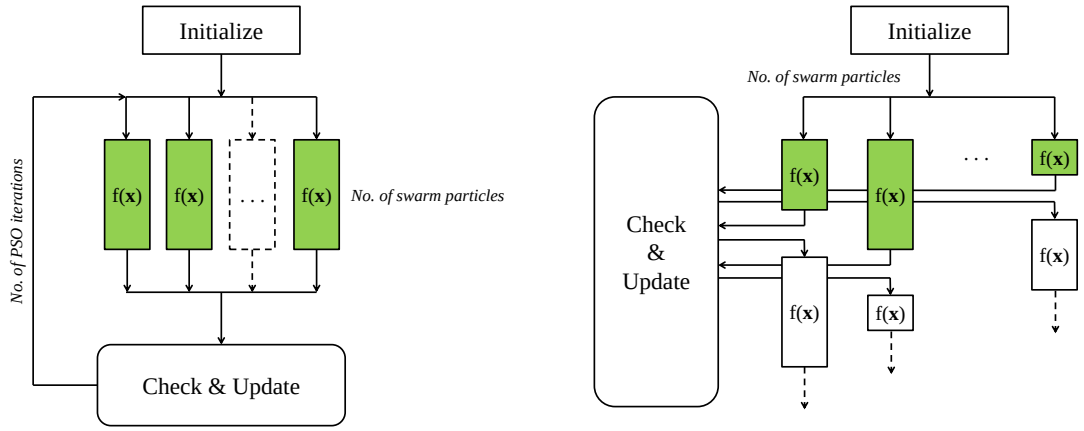


Figure 1.1: Block diagrams for parallel (a) S-DPSO and (b) A-DPSO: the green boxes represent the first set of particles evaluated by the algorithm

1.2 Hybrid global/local DPSO algorithm

In the framework of derivative-free optimization, combining heuristic procedures and exact methods could be amenable, provided that:

- the overall hybridized scheme is efficient, i.e. it is possibly not too expensive. A legitimate expectation is that the overall computational cost of the combined scheme is in-between the cost of (not combined) DPSO and the cost of the exact method;
- the results provided by the combined procedure are endowed with some theoretical properties, which are guaranteed by an effective combination of DPSO and the exact method. Typical theoretical properties characterize both the convergence of sequences of points, and the stationarity of limit points of the sequences generated by the hybridized scheme.

Herein, the focus is on some modifications of DPSO, where converging subsequences of iterates are generated. The modifications proposed for DPSO guarantee that the generated sequences of iterates have subsequences converging to stationary points of the objective function (see also [57, 58, 59, 60]). In particular, since there are in the literature theoretical results for several exact derivative-free methods [21, 61], DPSO is combined with a line search-based derivative-free algorithm, which still an unexplored issue, apart from the analysis in [22].

The aim is to provide robust methods to force the convergence of subsequences of points toward a stationary point, which satisfies first order optimality conditions for $f(\mathbf{x})$.

1.2.1 Line-search method

The last two decades have seen in particular the blow up of a remarkably effective class of optimization methods, endowed with complete convergence analysis and competitive performance: namely direct search methods. The latter class (see [21]) counts several optimization methods, which do not use derivatives but basically rely on "*the*

ranks of a countable set of function values" [21], i.e. on comparing the objective function values in specific points of the search space.

Among direct search methods, herein, the focus is on a subclass of iterative techniques, which is usually addressed in the literature as *Generating Set Search* (GSS). In the latter class, the main idea is that of decreasing the objective function at each iteration, on a *cone* in $\mathbb{R}^N$ generated by suitable search directions. *Pattern search methods* are in the GSS class, and have the distinguishing feature of enforcing, at each iteration, a simple decrease of the objective function. Conversely, also *line search-based* derivative-free methods are iterative schemes in GSS class, however they impose at each iteration a so called *sufficient reduction* of $f(\mathbf{x})$.

In literature, there is plenty of examples where evolutionary strategies are combined with GSS schemes and yield globally convergent algorithms [57, 62, 63]. In particular, in the last reference PSO is hybridized within a pattern search framework, and a resulting method converging to stationary points is given. Observe that in the literature of derivative-free methods it can be also find PSO-based approaches combined with a trust-region framework [62, 63], in order to provide again globally convergent methods to stationary points.

The following very preliminary results [61] can help the reader grasp the importance of the GSS class, in order to ensure convergence to stationary points.

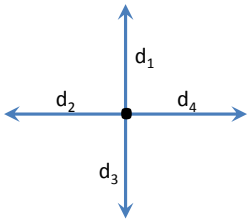
Given the set of vectors $D = \{d_1, \ldots, d_m\}$ of $\mathbb{R}^N$, D is defined as Positively Spanning Set (PSS) if for any vector $u \in \mathbb{R}^N$

$$u = \sum_{i=1}^{m} \alpha_i d_i, \qquad \alpha_i \geq 0 \tag{1.9}$$

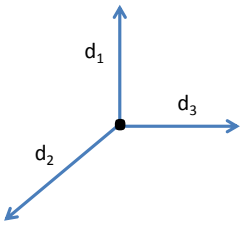
i.e. any vector u of $\mathbb{R}^N$ can be expressed as the weighted sum of the vectors in D, using nonnegative weights.

Thus, a PSS substantially provides a set of vectors which positively span the space $\mathbb{R}^N$. It can be easily proved that if D is a PSS of $\mathbb{R}^N$, then its cardinality must be at least $N+1$. It is very easy to define PSSs; simple examples of them in $\mathbb{R}^2$ are given in Fig. 4.3, where $m = 4$ (*top* and *bottom*) and $m = 3$ (*middle*).

$$D_{\oplus} = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$



$$D = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$



$$D = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

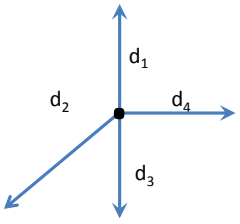


Figure 1.2: Examples of PSSs in $\mathbb{R}^2$. The subscript '$\oplus$' in the uppermost PSS means that the vectors in the set are the coordinate unit vectors $\pm e_i$, $i = 1, \ldots, N$.

In addition, there is the following nice property of PSSs: if the point $x \in \mathbb{R}^N$ is not stationary for f in Eq. 1.1 (i.e. $\nabla f(\mathbf{x}) \neq 0$), given the PSS D in $\mathbb{R}^N$, there exists at least one vector, say $\hat{d} \in D$, such that $\nabla f(\mathbf{x})^T \hat{d} < 0$, meaning that the direction $\hat{d}$ is of descent for $f(\mathbf{x})$ at $\mathbf{x}$. The latter fact ensures that if the current point is not stationary, and a PSS is available, roughly speaking there is at least one *direction of descent* for $f(\mathbf{x})$ in the PSS. Consequently, if the PSS D is available and the value $f(\mathbf{x})$ cannot be decreased on points along all the directions in D, then it means that the iterate $\mathbf{x}$ is a stationary point.

Detail of mathematical assumption and theorems can be found in [23].

1.2.2 LS-DF_PSO formulation

The hybrid algorithm, proposed herein, is obtained by coupling the DPSO scheme described in Sec. 1.1.2 with the derivative-free line-search method based on a PSS, where the set of search directions (D) is defined by the unit vectors $\pm e_i$, $i = 1,\dots,N$, as shown in the following equation

$$D = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} \tag{1.10}$$

This hybrid algorithm, namely LS-DF_PSO is endowed with both the local convergence properties of line-search algorithm and the global strategies of exploration of DPSO. The LS-DF_PSO algorithm is described in Alg. 3, where ς_0 is the initial step size for line-search, proportional to search space dimension, μ is a tolerance imposed to stop the local searches, ϑ is ratio of reduction of the step at each line-search iteration, and D_i is local searches direction vector as per Eq. 1.10.

Algorithm 3 LS-DF_PSO pseudo-code

```
Initialize a swarm of N_p particles
Set line-search step ς = ς_0 · (max(u) − min(l))
while (n < Max number of iterations) do
   for j = 1,N_p do
      Evaluate f(x_j^n)
   end for
   Update {x_{j,pb}}, x_gb
   Update particle positions and velocities {x_j^{n+1}}, {v_j^{n+1}}
   if x_gb is not improved, start the local searches then
      while ς > μ and (n < Max number of iterations) do
         for i = 1,2N do
            Evaluate f(x^n) in x_gb + ς · D_i
         end for
         if x_gb is improved then
            go to (4)
         else
            ς = ς · ϑ
         end if
      end while
   end if
end while
Output the best solution found
```

1.3 DIviding RECTangle algorithm

DIRECT is a sampling deterministic global derivative-free optimization algorithm and a modification of the Lipschitizian optimization method [45]. It starts the optimization by transforming the search domain $\mathcal{D}$ of the problem into the unit hyper-cube $\mathcal{U}$. At the first step of DIRECT, $f(\mathbf{x})$ is evaluated at the center (c) of $\mathcal{U}$; the hyper-cube is then partitioned into a set of smaller hyper-rectangles and $f(\mathbf{x})$ is evaluated at their centers. Let the partition of $\mathcal{U}$ at iteration n be defined as

$$\begin{aligned} &\mathcal{H}^n = \{\mathcal{U}_j : j \in \mathcal{J}^n\}, \text{ with} \\ &\mathcal{U}_j = \{\mathbf{x} \in \mathbb{R}^N : \ell_i^{(j)} \le x_i \le u_i^{(j)},\ i = 1,\dots,N,\ \forall j \in \mathcal{J}^n\} \end{aligned} \tag{1.11}$$

where N is the number of design variables, $\ell_i^{(j)}$ and $u_i^{(j)} \in [0,1]$, with $j \in \mathcal{J}^n$, are the lower and upper bounds defining the hyper-rectangle $\mathcal{U}_j$, and $\mathcal{J}^n$ is the set of indices identifying the subsets defining the current partition. At a generic n-th iteration of the algorithm, starting from the current partition $\mathcal{H}^n$ of $\mathcal{U}$, a new partition, $\mathcal{H}^{n+1}$, is built by subdividing a set of promising hyper-rectangles of the previous one. The identification of "potentially optimized" hyper-rectangles is based on some measure of the hyper-rectangle itself and on the value of $f(\mathbf{x})$ at its center c^j. The refinement of the partition continues until a prescribed number of function evaluations have been performed, or another stopping criterion is satisfied. The minimum of $f(\mathbf{x})$ over all the centers of the final partition, and the corresponding centers, provide an approximate solution to the problem. It may be noted that the box constraints are automatically satisfied.

Figure 1.3 shows several iterations of the DIRECT algorithm: each row represents a new iteration; the transition from the first column to the second represents the identifying process of the potentially optimal hyper-rectangles; the shaded rectangles in the second column are the potentially optimal hyper-rectangles as identified by DIRECT; the third column shows the domain after these potentially optimal rectangles have been divided.

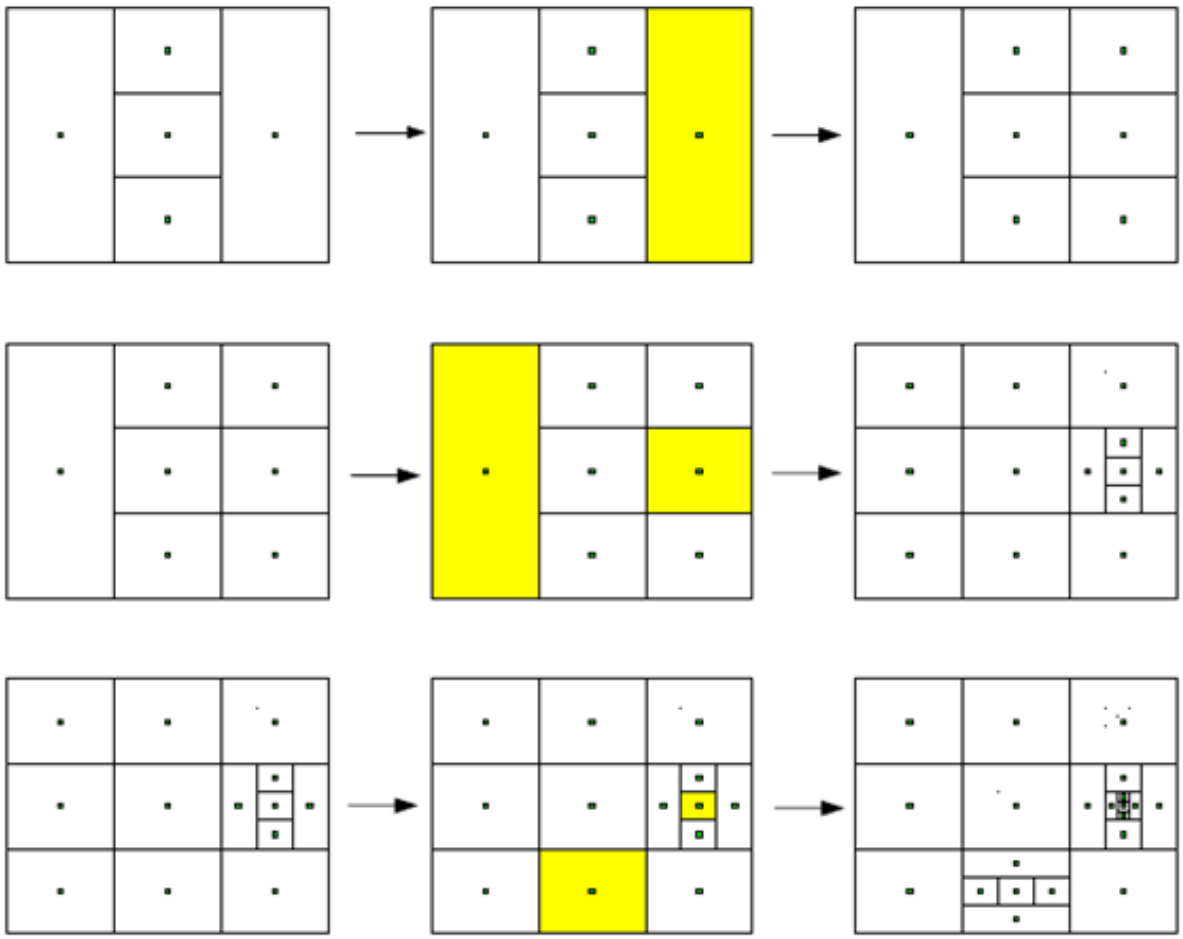


Figure 1.3: Sketch of DIRECT iterations [64]

The DIRECT algorithm is reported in Alg. 4 [65], where C^n are the centers of the hyper-rectangles $\mathcal{H}^n$ at the n-th iteration.

Algorithm 4 DIRECT pseudo-code

1: $\mathcal{H}^n = \{\mathcal{U}\}$, with $n = 0$, $\mathbf{c} =$ center of $\mathcal{U}$, $f_{\min} = f(\mathbf{c})$, $\mathbf{x}_{\min} = \{\mathbf{c}\}$, $N_{\max} \geq 0$
2: **repeat**
3: Set $n = n+1$, identify the set of indices $\mathcal{J}^n_\star \subseteq \mathcal{J}^n$ of the potentially optimal hyper-rectangles in $\mathcal{H}^n$
4: For each $j \in \mathcal{J}^n_\star$ subdivide $\mathcal{U}_j$ (generate the new partition $\mathcal{H}^{n+1}$
5: Evaluate f in the centers of the new hyper-rectangles
6: $f_{\min} = \min\{f(\mathbf{c}) : \mathbf{c} \in C^n\}$, $\mathbf{x}_{\min} \in \{\mathbf{c} \in C^n : f(\mathbf{c}) = f_{\min}\}$
7: **until** function evaluations $> N_{\max}$
8: **return** $f_{\min}, \mathbf{x}_{\min}$

1.4 Hybrid global/local DIRECT-type algorithm

The original DIRECT algorithm has some known weaknesses. First of all, in practice, it is hard to develop some efficient stopping condition other than, e.g., exhaustion of some preset computational resources. Furthermore, DIRECT is typically quite fast in getting close to the global optimum, but it can be slow to converge to the solution

with an high accuracy. In fact, it can often waste a lot of time exploring uninteresting regions of the feasible domain, thereby delaying the discovery of global minima.

Various attempts have been done in the literature to overcome this latter source of inefficiency. For example, different techniques for adaptive setting the DIRECT balancing parameter can be applied as reviewed [66]; transformations of the search domain can be considered [67, 68, 69]; smart schemes for balancing local and global information within the same global optimization procedure can be introduced [66, 70, 71, 72]; hybridization of a DIRECT-type method by cleverly invoking separate local optimizers during the global optimization process is also very useful [67, 73, 74]. The two latter approaches are particularly beneficial for solving multiextremal large-scale problems.

In the following two local hybridization by derivative-free line-search technique are presented.

1.4.1 DIRMIN formulation

DIRMIN is a hybridization of the DIRECT algorithm with a derivative-free local search algorithm. The derivative-free local searches are performed starting form the centers c^j of the "potentially optimal" hyper-rectangles identified by DIRECT methods. The DIRMIN algorithm, recalled from [65], is reported in Alg. 5.

Algorithm 5 DIRMIN pseudo-code

1: $\mathcal{H}^n = \{\mathcal{U}\}$, with $n = 0$, $\mathbf{c}$ = center of $\mathcal{U}$, $f_{\min} = f(\mathbf{c})$, $\mathbf{x}_{\min} = \{\mathbf{c}\}$, $\beta > 0$, $\gamma \in [0,1]$, $N_{\max} \geq 0$
2: **repeat**
3: Set $n = n+1$, identify the potentially optimal hyper-rectangles $\mathcal{P}^n$ in $\mathcal{H}^{n-1}$ and set $f_{\text{mold}} = f_{\min}$
4: **if** function evaluations $\geq \gamma \cdot N_{\max}$ **then**
5: for all centroids $\mathbf{c}^j$ of hyper-rectangles in $\mathcal{P}^n$ perform a local minimization until the maximum step-length becomes smaller than the tolerance β, and record the best function value f_{ml}
6: **else**
7: set $f_{\text{ml}} = +\infty$
8: **end if**
9: subdivide the potentially optimal hyper-rectangles to build $\mathcal{H}^n$
10: evaluate f at the centers of the new hyper-rectangles
11: $f_{\min} = \min\left\{f_{\text{mold}}, f_{\text{ml}}, \min\{f(\mathbf{c}) : \mathbf{c} \in C^n\}\right\}$, $\mathbf{x}_{\min} \in \{\mathbf{x} \in \mathcal{U} : f(\mathbf{x}) = f_{\min}\}$, where C^n is the set of centroids $\mathbf{c}$ of the hyper-rectangles in $\mathcal{H}^n$
12: **until** function evaluations $> N_{\max}$
13: **return** $f_{\min}$, $\mathbf{x}_{\min}$

Here, $\mathcal{U}$ represents the unit hyper-cube, β is the tolerance used in the stopping criterion of the derivative-free local minimizations, γ is the activation trigger defining the starting point of the derivative-free local searches as ratio of the maximum number of function evaluation $N_{\max}$, and f_{ml} is the minimum value found by the derivative-free local searches. At each iteration n, every hyper-rectangle in $\mathcal{H}^n$ is characterized by the length of its diagonal and the value of the objective function at its centroid. Hence, provided that the Lipschitz constant is known, for every hyper-rectangle it is possible to compute a lower bound. An hyper-rectangle is declared potentially optimal ($\mathcal{P}^n$) and then selected for further subdivision if an estimate $L > 0$ of the Lipschitz constant exists such that it yields the best estimated lower bound among all the hyper-rectangles. The subdivision performed in Step (9) is carried out by dividing the hyper-rectangles along the longest edges, thus guaranteeing that the hyper-rectangles shrink on every dimension in a sufficiently balanced way.

The local minimizations at Step (4) are performed by using the derivative-free local optimization algorithm for bound constrained problems proposed in Ref. [75]. It performs derivative-free line searches along the coordinate directions. At every iteration, the maximum of the step-lengths gives a measure of stationarity of the current iterate (see e.g. [21]) and motivates the stopping criterion adopted at Step (4) of DIRMIN. As shown by [65], the DIRMIN algorithm can be efficient, in terms of function evaluations, with respect to the original DIRECT algorithm. The maximum number of function evaluations is used as stopping criterion of DIRMIN. It should be noted that the DIRECT algorithm is obtained from DIRMIN by simply replacing Step (4) with the assignment $f_{\text{ml}} = +\infty$.

1.4.2 DIRMIN-2 formulation

DIRMIN-2 is a modification of DIRMIN. Rather than performing the derivative-free local minimizations starting from the centroids of all the potentially optimal hyper-rectangles $\mathcal{P}^n$, a single derivative-free local minimization is performed starting from the best point produced by dividing the potentially optimal hyper-rectangles. The derivative-free local optimization algorithm proposed in Ref. [75] is used. Details of DIRMIN-2 are given Alg. 6

Algorithm 6 DIRMIN-2 pseudo-code

1: $\mathcal{H}^n = \{\mathcal{U}\}$, with $n = 0$, $\mathbf{c} =$ center of $\mathcal{U}$, $f_{\min} = f(\mathbf{c})$, $\mathbf{x}_{\min} = \{\mathbf{c}\}$, $\beta > 0$, $\gamma \in [0,1]$, $N_{\max} \geq 0$
2: **repeat**
3: Set $n = n+1$, identify the potentially optimal hyper-rectangles $\mathcal{P}^n$ in $\mathcal{H}^{n-1}$ and set $f_{\text{mold}} = f_{\min}$
4: subdivide the potentially optimal hyper-rectangles to build $\mathcal{H}^n$
5: evaluate f at the centers of the new hyper-rectangles
6: let $\tilde{\mathbf{c}} \in \arg\min\{f(\mathbf{c}) : \mathbf{c} \in C^n\}$, where C^n the set of the centers of the hyper-rectangles $\mathcal{H}^n$
7: **if** function evaluations $\geq \gamma \cdot N_{\max}$ **then**
8: perform a local minimization starting from $\tilde{\mathbf{c}}$ until the maximum step-length becomes smaller than the tolerance β, and let f_{ml} be the best function value found
9: **else**
10: set $f_{ml} = +\infty$
11: **end if**
12: $f_{\min} = \min\left\{f_{\text{mold}}, f_{\text{ml}}, \min\{f(\mathbf{c}) : \mathbf{c} \in C^n\}\right\}$, $\mathbf{x}_{\min} \in \{\mathbf{x} \in \mathcal{U} : f(\mathbf{x}) = f_{\min}\}$
13: **until** function evaluations $> N_{\max}$
14: **return** $f_{\min}$, $\mathbf{x}_{\min}$

Note that, when $\arg\min\{f(\mathbf{c}) : \mathbf{c} \in C^n\}$ is not a singleton, $\tilde{\mathbf{c}}$ is the centroid of the first hyperectangle (among those produced at step 4) for which $f(\tilde{\mathbf{c}}) = \min\{f(\mathbf{c}) : \mathbf{c} \in C^n\}$. The rationale behind the definition of DIRMIN-2 hinges on considering the subdivision of potentially optimal hyper-rectangles as a crude kind of local search, which can be improved by the use of a more sophisticated and efficient local minimization algorithm.

Also in this case, the DIRECT algorithm is obtained from DIRMIN-2 by simply replacing Step (6) and (7) with the assignment $f_{\text{ml}} = +\infty$.

1.5 Fish Shoal Algorithm

The fish shoal algorithm (FSA) is a deterministic derivative-free global optimization algorithm, introduced and developed in [46, 76], for solving engineering optimization problems with costly objective functions. The method is intended for unconstrained single-objective maximization and is based on a simplified social model of a fish shoal in search for food. FSA is formulated starting from the dynamics of a single individual belonging to a fish shoal in search for food, and subject to a shoal attraction force and a food-related attraction force. This formulation differentiates FSA from the artificial fish-swarm algorithm [77] and other metaheuristics such as PSO, for the direct use of the objective function value (food-related attraction force).

1.5.1 A brief overview of fish shoal behavior

In biology, fishes that stay in group for social reasons are shoaling. Shoaling offers numerous benefits to individual fish, including increased success in finding food, access to potential mates, and increased protection from predators. Fish shoals might be a group of mixed species and sizes that have gathered randomly near some local resource. Although shoaling fish can relate to each other in a loose way, with each fish swimming and foraging somewhat independently, they are nonetheless aware of the other members of the group as shown by the way they adjust behavior such as swimming, so as to remain close to the other fish in the group. If the shoal becomes more tightly organised, with the fish synchronising their swimming so they all move at the same speed and in the same direction in a coordinated manner, they are schooling. Shoaling fish can shift into a disciplined and coordinated school, then shift back to an amorphous shoal within seconds. Such shifts are triggered by changes of activity from feeding, resting, travelling or avoiding predators.

Pitcher et al. [78], in their study of foraging behavior in shoaling cyprinids, have proven that swimming in groups enhances foraging success. In this study, the time it took for groups of minnows and goldfish to find a patch of food was quantified. The number of fishes in the groups was varied, and a statistically significant decrease in the amount of time necessary for larger groups to find food was established. This is directly related to the presence of many eyes searching for the food, because fishes in shoal share information by monitoring each others behavior closely. Feeding behavior in one fish quickly stimulates food-searching behavior in others [79]. Observations on the foraging behavior of captive golden shiner found they formed shoals, which were led by a small number of experienced individuals who knew when and where food was available [80]. If all golden shiners in a shoal have similar knowledge of food availability, there are few individuals that still emerge as natural leaders and behavioral tests suggest they are naturally bolder. Fish generally prefer larger shoals [81]. Larger shoals may find food

faster, though that food would have to be shared amongst more individuals. Competition may mean that hungry individuals might prefer smaller shoals or exhibit a less preference for very large shoals.

Such foraging behavior of fish shoal can be formulated in such a way that it can be associated with the objective function to be optimized, and this makes possible to formulate novel optimization algorithms.

1.5.2 FSA formulation

Consider an optimization problem of the type

$$\begin{array}{rl} \text{Maximize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{array} \tag{1.12}$$

where f is the objective function, $\mathbf{x} \in \mathbb{R}^N$ is the variable vector $\mathcal{D}$ normalized into a unit hyper-cuber $\mathcal{U}$, with lower and upper bounds equal to $\mathbf{l}$ and $\mathbf{u}$ respectively.

Now consider a foraging shoal of individuals $\mathbf{x}_j$, exploring the research space with the aim of finding an approximate solution for problem in Eq. 1.12. The shoal is modelled as a dynamical system and the dynamics of the j-th individual depends on a shoal attraction force $\boldsymbol{\delta}_j$ (Fig. 1.4a), and a food-related attraction force $\boldsymbol{\varphi}_j$ (Fig. 1.4b), as

$$m\ddot{\mathbf{x}}_j + \xi\dot{\mathbf{x}}_j + k\boldsymbol{\delta}_j = h\boldsymbol{\varphi}_j \tag{1.13}$$

where

$$\boldsymbol{\delta}_j = -\sum_{i=1}^{N_s} (\mathbf{x}_i - \mathbf{x}_j) \tag{1.14}$$

and

$$\boldsymbol{\varphi}_j = \sum_{i=1}^{N_s} \frac{2\Delta f(\mathbf{b}_i, \mathbf{x}_j)}{1 + \|\mathbf{b}_i - \mathbf{x}_j\|^{\alpha}} \mathbf{e}(\mathbf{b}_i, \mathbf{x}_j) \tag{1.15}$$

with

$$\Delta f(\mathbf{b}_i, \mathbf{x}_j) = \frac{f(\mathbf{b}_i) - f(\mathbf{x}_j)}{\rho}, \qquad \mathbf{e} = \frac{\mathbf{b}_i - \mathbf{x}_j}{\|\mathbf{b}_i - \mathbf{x}_j\|} \tag{1.16}$$

In the above equations, m, ξ, k and $h \in \mathbb{R}^+$ define the shoal dynamics; $N_s \in \mathbb{N}^+$ is the shoal size; $\alpha \in \mathbb{R}^+$ tunes the food-related attraction force (see Fig. 1.4b); $\mathbf{x}_j \in \mathbb{R}^N$ is the vector-valued position of the j-th individual; $f(\mathbf{x}) \in \mathbb{R}$ is the objective function (representing the food distribution); $\mathbf{b}_i$ is the best position ever visited by the i-th individual; $\rho = f(\mathbf{b}) - f(\mathbf{w}^n)$ is a dynamic normalization term for f, with $\mathbf{b}$ the best position ever visited by the shoal and $\mathbf{w}^n$ the worst position occupied by the shoal individuals at the current iteration n.

Using the explicit Euler integration scheme, the FSA iteration become

$$m\frac{\mathbf{v}_j^{n+1} - \mathbf{v}_j^n}{\Delta t} = -\xi\,\mathbf{v}_j^n - k\boldsymbol{\delta}_j + h\boldsymbol{\varphi}_j \tag{1.17}$$

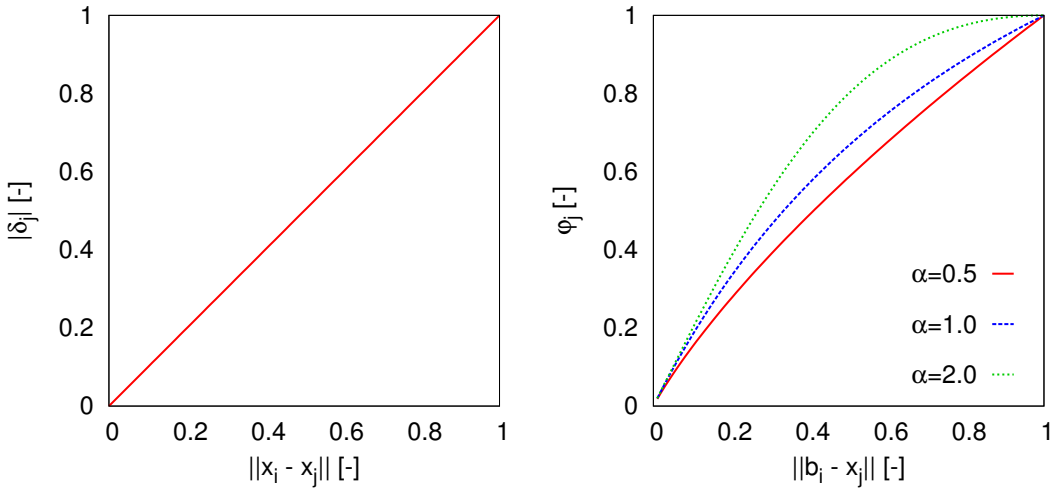
which yields

$$\mathbf{v}_j^{n+1} = \mathbf{v}_j^n + \frac{\Delta t}{m}\left(-\xi\mathbf{v}_j^n - k\boldsymbol{\delta}_j + h\boldsymbol{\varphi}_j\right) \tag{1.18}$$

Setting $m = 1$ finally gives

$$\begin{cases} \mathbf{v}_j^{n+1} = (1 - \xi\Delta t)\mathbf{v}_j^n + \Delta t(-k\boldsymbol{\delta}_j + h\boldsymbol{\varphi}_j) \\ \mathbf{x}_j^{n+1} = \mathbf{x}_j^n + \mathbf{v}_j^{n+1}\Delta t \end{cases} \tag{1.19}$$

where $\mathbf{x}_j^n$ and $\mathbf{v}_j^n$ represent the individual position and velocity vector of the j-th individual at the n-th iteration, respectively.



(a) (b)

Figure 1.4: Shoal (a) and food-related (b) attraction forces, for $N_s = 2$ and $\Delta f = 1$

In Eq. 1.19, the integration step Δt must guarantee the stability of the explicit Euler scheme, at least for the free dynamics. To this aim, consider the free dynamics of the k-th component of $\mathbf{x}$ (or k-th variable), say a. Consider the dynamics of a for the j-th individual

$$\ddot{a}_j + \xi \dot{a}_j + k\delta_j = 0 \tag{1.20}$$

and finally for the entire shoal

$$\begin{pmatrix} \dot{\mathbf{a}} \\ \dot{\mathbf{c}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{c} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{a} \\ \mathbf{c} \end{pmatrix} \tag{1.21}$$

where

$$\mathbf{K} = -k \begin{pmatrix} N_s - 1 & -1 & -1 & \cdots & -1 \\ -1 & N_s - 1 & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & \cdots & -1 & N_s - 1 & -1 \\ -1 & \cdots & -1 & -1 & N_s - 1 \end{pmatrix}, \qquad \mathbf{G} = \xi \mathbf{I} \tag{1.22}$$

and $\mathbf{I}$ is the $[N_s \times N_s]$ identity matrix.

The solution of Eq. 1.21 is stable if

$$Re(\lambda) \leq 0 \tag{1.23}$$

where $\lambda = -\gamma \pm i\omega$ are eigenvalues of $\mathbf{A}$. This yields

$$\Delta t \leq \left. \frac{2\gamma}{\gamma^2 + \omega^2} \right|_{\min} = \Delta t_{\max} \tag{1.24}$$

Equation 1.19 represents a fully informed formulation, where each individual knows the story of the whole shoal. The FSA pseudo-code is shown in Alg. 7.

Algorithm 7 FSA pseudo-code

1: Normalize $\mathbf{x}$ into a unit hypercube $\mathcal{U}$
2: Initialize a shoal of N_s individuals
3: Evaluate the $\Delta t_{\max}$ by the linear system's eigenvalues (Eq. 1.21)
4: **while** $n <$ Max number of iterations **do**
5: **for** $j = 1, N_s$ **do**
6: Evaluate $f(\mathbf{x}_j)$
7: **end for**
8: Update $\mathbf{x}_{j_b}$, $f(\mathbf{x}_{j_b})$ and the attraction forces
9: Update $\mathbf{v}_j^{n+1}$ and $\mathbf{x}_j^{n+1}$
10: **end while**
11: Output the best solution found

Chapter 2

Shape design modification

"I – I hardly know, sir, just at present – at least I know who I WAS when I got up this morning, but I think I must have been changed several times since then."

— Lewis Carroll, *Alice in Wonderland*

Shape design optimization finds the optimum shape for a given structural layout [82]. Obviously, the choice of the shape parametrization technique has a large impact on the practical implementation and often also on the success of the optimization process. Shape deformation methods have been an area of continuous and extensive research within the fields of computer graphics and geometry modelling. Consequently, a wide variety of techniques has been proposed during recent years [83]. The parametrization techniques can be divided into the following categories [82]: basis vector, domain element, partial different equation, discrete, polynomial and spline, CAD-based, analytical, and free-form deformation (FFD). In the context of SBDO, to be successful, the parametrization model must yield a compact and effective set of design variables so the solution time would be feasible. As a general statement it can be said that, in principle, the number (and the type) of parameters implicitly defines the diversity of the admissible shapes: since the larger the variety of potential designs, the larger the improvements it can be hope to find (starting from the original design), the importance of a proper choice of the design parameters is evident [84].

Starting from an initial design, so that all the details of the original shape are known, the adaptation of an existing surface/volumetric simulation grid according to an updated CAD geometry is a key component for performing the optimization in a fully automatic procedure. The importance of such a component further increase when dealing with complex geometry that prohibit automatic mesh generation or when using population-based optimization algorithm, i.e. evolutionary algorithm, which typically require the creation and evaluation of a large number of design variations in order to find a feasible solution [85]. Another issue that has to be addressed is that the computational grid adopted in the analysis must be regenerated or deformed each time there is the need to evaluate a new perturbed design, and this operation has to be performed in background, without any guarantee about the quality of the new mesh. When this has to be done in conjunction with CFD solvers, the regridding issues may become extremely relevant to the performance and the final result of the optimization [84]. In order to avoid the costly mesh generation process for each design variation created during the optimization, one typically aims at adapting an initial simulation mesh alongside with the surface.

A type of deformation methods that naturally fulfils the above requirements are so-called space deformations. Deformation (more than regeneration) of the computational grid represents a good approach to the problem: (1) one doesn't need to regenerate the whole volume grid each time the hull shape is perturbed, (2) the initial hull shape is preserved and (3) one can deform some part of the hull with a prescribed degree of continuity. The fundamental idea behind these methods is to deform the embedding space around an object, thereby deforming the object implicitly. From a mathematical point of view a space deformation is a function

$$\boldsymbol{\delta}_s : \mathbb{R}^3 \longrightarrow \mathbb{R}^3 \tag{2.1}$$

that maps each point in space to a certain displacement. Given a deformation function, a geometry $\mathcal{G}$ can be transformed to a deformed geometry $\mathcal{G}'$ by computing updated point locations

$$\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}_s(\mathbf{x}) \tag{2.2}$$

for each original point $\mathbf{x} \in \mathcal{G}$ [83].

The following subsections show three different shape modification technique used in the present work, that satisfied the requirements described above. Specifically, free-form deformation (FFD), and orthogonal expansion over 2D and 3D subdomains are presented and described for using in simulation-based shape design optimization in naval engineering.

Finally, since the use of the present (and also other) techniques allows the use of a possibly infinite number of variables causing an increase in the optimization costs, a design-space dimensionality reduction technique based on generalized Karhunen-Loève expansion [16, 86] is presented, in order to define a more effective and efficient (from the optimization point of view) design modification space.

2.1 Free-form deformation

Free-form deformation (FFD) is a geometry deformation technique used to model simple deformations of rigid objects, widely used in both academia and industry for SBDO [16, 47, 84, 87]. The technique was first described in [88], and is based on an earlier technique described in [89]. It is based on the idea of embedding an object to be deformed within a parallelepiped/cube lattice (see Fig. 2.1) or another shell object, and transforming the object within the cube as the lattice is deformed. Deformation of the lattice is based on the concept of so-called hyper-patches, which are three-dimensional analogues of parametric curves such as Bézier curves, B-Splines, or NURBS (Non Uniform Rational Basis-Splines). The deformation procedure can be divided in several step: (1) a control lattice point has to be generated and adapted to the deformation scenario, (2) the local coordinates with respect to the control lattice have to be computed for each point to be deformed, (3) express each object point $\mathbf{x} \in \mathcal{G}$ as a linear combination of lattice control points $\mathbf{c}_{ijk}$ and basis function B_i so that

$$\mathbf{x} = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} \mathbf{c}_{ijk} B_i(u) B_j(v) B_k(w) \tag{2.3}$$

where (u,v,w) are the local coordinates of $\mathbf{x}$ with respect to the control lattice, and l, m, n are the numbers of control points in each direction. Defining

$$\mathbf{u}(\mathbf{x}) := (u,v,w), \qquad B_p(\mathbf{u}(\mathbf{x})) := B_i(u) B_j(v) B_k(w) \tag{2.4}$$

as well as

$$\boldsymbol{\delta}\mathbf{c}_p := \boldsymbol{\delta}\mathbf{c}_{ijk} = \mathbf{c}'_{ijk} - \mathbf{c}_{ijk} \tag{2.5}$$

where $\mathbf{c}'_{ijk}$ denotes an updated control point location, the FFD space deformation function can be expressed by

$$\boldsymbol{\delta}_s(\mathbf{x}) = \sum_{p} \boldsymbol{\delta}\mathbf{c}_p B_p(\mathbf{u}(\mathbf{x})) \tag{2.6}$$

Finally, the deformation is performed by moving the control point and computing the updated object point locations.

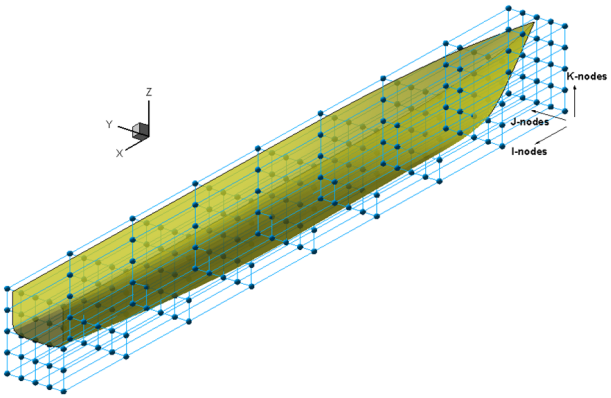


Figure 2.1: Example of FFD parallelepiped lattice on the Delft catamaran demi-hull [16]

The FFD formulation is independent of grid topology and that independence makes it suitable for a variety of analysis codes, such as low- and high-fidelity analysis tool. On the other hand, the FFD variables may have little or no physical significance for the design engineers, thereby making it difficult to establish an effective and compact set of design variables.

2.2 Orthogonal basis functions expansion over 2D subdomains

An effective and efficient shape design modification method in the context of SBDO is introduced in [90]. The shape deformation is performed by the superposition, on the original shape/grid, of bi-dimensional orthogonal basis functions expansion (OBFE_2D). The orthogonal basis functions are linearly independent and are directly defined on the curvilinear coordinates ξ and η of the object (computational grid)

$$\boldsymbol{\psi}_i(\xi,\eta) \; : \; \mathcal{G} = [0, L_\xi] \times [0, L_\eta] \in \mathbb{R}^2 \longrightarrow \mathbb{R}^3, \qquad i = 1, ..., N \tag{2.7}$$

as

$$\boldsymbol{\delta}_s(\xi,\eta) = \sum_{i=1}^{N} \alpha_i \, \boldsymbol{\psi}_i(\xi,\eta) \tag{2.8}$$

where the coefficients $\alpha_i \in \mathbb{R}$ $(i = 1, \ldots, N)$ are the design variables,

$$\boldsymbol{\psi}_i(\xi,\eta) := \sin\left(\frac{r_i \pi \xi}{L_\xi} + \phi_i\right) \sin\left(\frac{t_i \pi \eta}{L_\eta} + \chi_i\right) \mathbf{e}_{q(i)} \tag{2.9}$$

and the following orthogonality property is imposed:

$$\iint_{\mathcal{G}} \boldsymbol{\psi}_i(\xi,\eta) \cdot \boldsymbol{\psi}_j(\xi,\eta) d\xi d\eta = \delta_{ij} \tag{2.10}$$

In Eq. 2.9, r_i and $t_i \in \mathbb{R}$ define the order of the function in ξ and η direction respectively; ϕ_i and $\chi_i \in \mathbb{R}$ are the corresponding spatial phases; L_ξ and $L_\eta \in \mathbb{R}$ define the domain size; $\mathbf{e}_{q(i)}$ is a unit vector, so that the design modifications may be applied in x, y or z direction, with $q(i) = 1$, 2, or 3 respectively. An example of OBFE_2D is shown in Fig. 2.2.

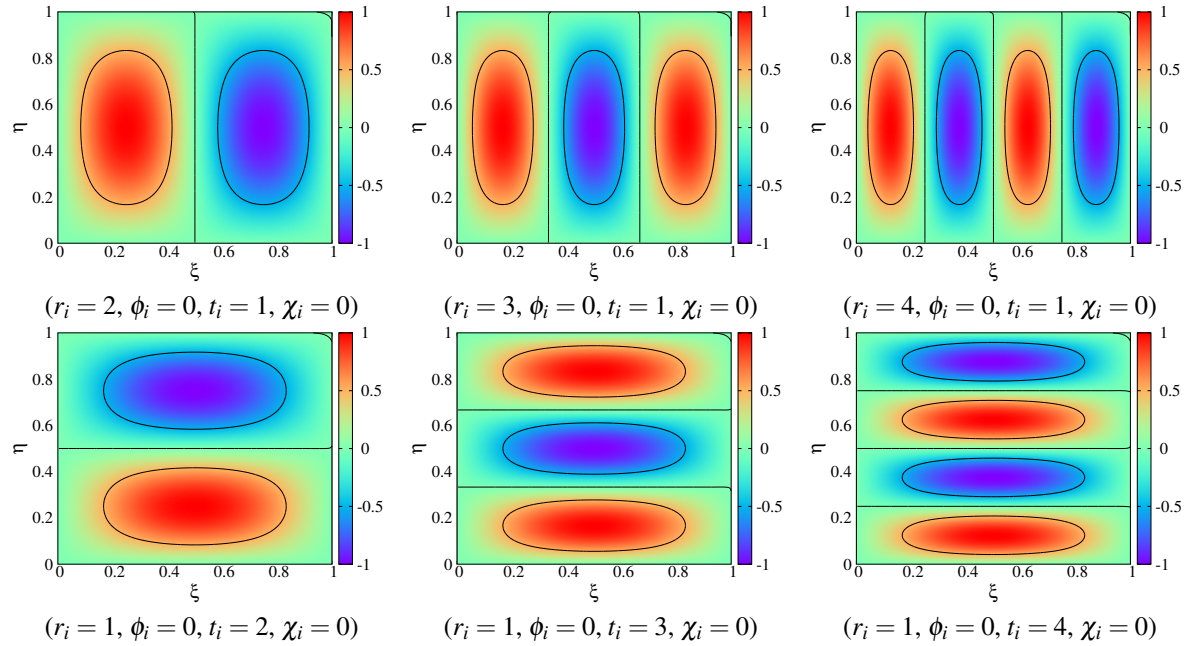


Figure 2.2: Example of 2D orthogonal functions $\boldsymbol{\psi}_i(\xi,\eta)$

This kind of technique can be applied on the whole object or a part of it, and the OBFE_2D design variables can be related to a physical meaning, e.g. Fig. 2.2a shows the movement of volumes from a side to another of the object conditional to the zone of application and the direction of the patch. On the other hand, the limit of this approach is in the direct use of the curvilinear coordinates corresponding to the object grid: the same orthogonal function corresponds to different shape modifications, dependently to grid discretization.

2.3 Orthogonal basis functions expansion over 3D subdomains

The OBFE_2D method has been extended to three-dimensional orthogonal basis functions expansion (OBFE_3D) [91], to release the shape modification method from the computational grid topology and to be more effective and efficient in the context of SBDO. The shape deformation is performed by the superposition, on the original shape/grid, of OBFE_3D defined on the Cartesian coordinates x, y, z over a hyper-rectangle

$$\boldsymbol{\varphi}_i(x,y,z) \; : \; \mathcal{G} = [0,L_x] \times [0,L_y] \times [0,L_z] \in \mathbb{R}^3 \longrightarrow \mathbb{R}^3, \qquad i = 1,...,N \tag{2.11}$$

as

$$\boldsymbol{\delta}_s(x,y,z) = \sum_{i=1}^{N} \beta_i \, \boldsymbol{\varphi}_i(x,y,z) \tag{2.12}$$

where the coefficients $\beta_i \in \mathbb{R}$ $(i = 1,\ldots,N)$ are the design variables,

$$\boldsymbol{\varphi}_i(x,y,z) := \sin\left(\frac{n_i \pi x}{L_x} + \phi_i\right) \sin\left(\frac{m_i \pi y}{L_y} + \chi_i\right) \sin\left(\frac{l_i \pi z}{L_z} + \theta_i\right) \mathbf{e}_{q(i)} \tag{2.13}$$

and the following orthogonality property is imposed:

$$\iiint_{\mathcal{G}} \boldsymbol{\varphi}_i(x,y,z) \cdot \boldsymbol{\varphi}_j(x,y,z) dx dy dz = \delta_{ij} \tag{2.14}$$

In Eq. 2.13, n_i, m_i and $l_i \in \mathbb{R}$ define the order of the function in x, y and z direction respectively; ϕ_i, χ_i and $\theta_i \in \mathbb{R}$ are the corresponding spatial phases; L_x, L_y and $L_z \in \mathbb{R}$ define the hyper-rectangle dimensions; $\mathbf{e}_{q(i)}$ is a unit vector. Modifications may be applied in x, y or z direction, with $q(i) = 1$, 2, or 3 respectively.

This kind of method can be applied on the whole object or a part of it, and it is independent of grid topology. On the other hand, there isn't a direct physical meaning of the object surface modification because the OBFE_3D are non linear on the object surface.

2.4 Automatic grid modification

The shape modification, performed by the techniques previously presented, can be directly reflected in the body surface grid deformation, for this reason they are also suitable in the optimization problems involving solvers that need only a surface grid (e.g. potential flow). On the contrary, when the optimization problem involves volume grid solvers (e.g. RANS or FEM), the surface grid modifications need to be transposed also to the volume grid: the methodology used in the present work is presented in the following.

2.4.1 Volume grid modification

The volume grid (e.g. boundary layer grid) is automatically modified, in order to reflect the shape modification applied to the the body (surface) grid. Assume that the body surface grid is defined with index $J = 1$ and spanned by indices $I = 1,\ldots,I_{\max}$ and $K = 1,\ldots,K_{\max}$. Accordingly, the volume grid is spanned by $I = 1,\ldots,I_{\max}$, $J = 1,\ldots,J_{\max}$, and $K = 1,\ldots,K_{\max}$, with $J = J_{\max}$ corresponding to the outer surface. Once the grid nodes of the body surface at $J = 1$ are modified following the shape modification vector $\boldsymbol{\delta}_s$ (e.g. defined by FFD, OBFE_2D, or OBFE_3D), any arbitrary inner node of the volume grid ($J = 2,\ldots,J_{\max}-1$) is modified similarly to Eq. 2.2, as

$$\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta} \tag{2.15}$$

with

$$\boldsymbol{\delta} = \frac{l^* - l}{l^*} \boldsymbol{\delta}_s + \frac{l}{l^*} \boldsymbol{\delta}_s^* \tag{2.16}$$

where l is the distance between (original) inner and body surface nodes, with arbitrary J and $J = 1$ respectively (and same I and K indices); l^* is the distance between (original) outer and body surface nodes, with $J = J_{\max}$ and $J = 1$ respectively (and same I and K indices); $\boldsymbol{\delta}_s^*$ is the modification of the outer surface ($J = J_{\max}$):

$$\boldsymbol{\delta}_s^* = c \, \boldsymbol{\delta}_s \tag{2.17}$$

with $c \in \mathbb{R}_0^+$.

The distance l (and l^*) may be evaluated in the simplest form as the Euclidean distance $\bar{l}$:

$$l = \bar{l} = \|\mathbf{x}_0 - \mathbf{x}_{s,0}\| \tag{2.18}$$

Alternatively, the approximate curvilinear distance $\hat{l}$ along the grid line at constant I and K may be used [92]:

$$l = \hat{l} = \sum_{j=1}^{J-1} \|\mathbf{x}_0^{(j+1)} - \mathbf{x}_0^{(j)}\| \tag{2.19}$$

where superscripts indicate grid indices, limited to J for the sake of compactness (since I and K are constant).

2.5 Design-space dimensionality reduction by generalized Karhunen-Loève expansion

Shape optimization research has focused on shape and topology parameterizations, as a critical issue to achieve the desired level of design variability [93, 94, 95]. Some recent research focused on research space variability and dimensionality reduction for efficient analysis and optimization procedures [5, 25]. A quantitative approach based on the Karhunen-Loève expansion (KLE, also known as proper orthogonal decomposition, POD) has been formulated for a pre-optimization assessment of the shape modification variability (see the left box of Fig. 2.3) and the definition of a reduced-dimensionality global model of the design space [16, 96, 97]. The mathematical properties of the dimensionality-reduction by KLE are described in the following.

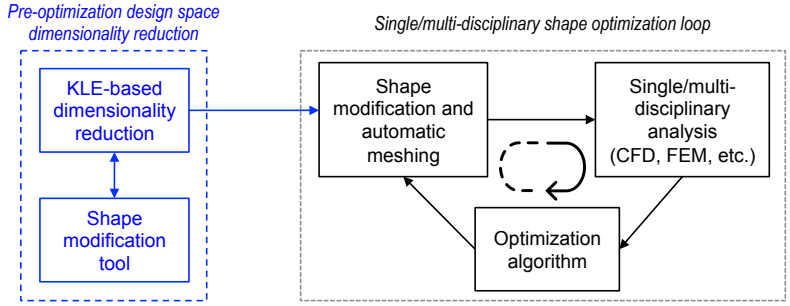


Figure 2.3: Scheme for single/multi-disciplinary shape optimization, including pre-optimization design-space dimensionality reduction

Consider a geometric domain of interest $\mathcal{G}$, which identifies the initial shape, and a set of coordinates $\mathbf{x} \in \mathcal{G}$. Assume that $\mathbf{u} \in \mathcal{U}$ is the design variable vector, which defines a shape modification vector $\boldsymbol{\delta}_s$. Consider the vector space of all possible square-integrable modifications of the initial shape, $\boldsymbol{\delta}_s(\mathbf{x},\mathbf{u}) \in L^2_\rho(\mathcal{G})$, where $L^2_\rho(\mathcal{G})$ is the Hilbert space defined by a generalized inner product

$$(\mathbf{f},\mathbf{g})_\rho = \int_{\mathcal{G}} \rho(\mathbf{x})\mathbf{f}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x}) d\mathbf{x} \tag{2.20}$$

with associated norm $\|\mathbf{f}\|_\rho = (\mathbf{f},\mathbf{f})_\rho^{\frac{1}{2}}$, where $\rho(\mathbf{x}) \in \mathbb{R}$ is a weight function. Generally, $\mathbf{x} \in \mathbb{R}^m$ with $m = 1,2,3$, $\mathbf{u} \in \mathbb{R}^N$ with N number of design variables, and $\boldsymbol{\delta}_s \in \mathbb{R}^l$ with $l = 1,2,3$ (with l not necessarily equal to m). Consider the design vector $\mathbf{u}$ as belonging to a stochastic space $\mathcal{S}$ with associated probability density function $f(\mathbf{u})$. The associated mean shape modification is evaluated as

$$\langle \boldsymbol{\delta}_s \rangle = \int_{\mathcal{U}} \rho(\mathbf{x}) \boldsymbol{\delta}_s(\mathbf{x},\mathbf{u}) f(\mathbf{u}) d\mathbf{u} \tag{2.21}$$

whereas the variance associated to the shape modification vector (geometric variance) is defined as

$$\sigma^2 = \left\langle \|\hat{\boldsymbol{\delta}}_s\|^2 \right\rangle = \int_{\mathcal{U}} \int_{\mathcal{G}} \rho(\mathbf{x}) \hat{\boldsymbol{\delta}}_s(\mathbf{x},\mathbf{u}) \cdot \hat{\boldsymbol{\delta}}_s(\mathbf{x},\mathbf{u}) f(\mathbf{u}) d\mathbf{x} d\mathbf{u} \tag{2.22}$$

where $\hat{\boldsymbol{\delta}}_s = \boldsymbol{\delta}_s - \langle \boldsymbol{\delta}_s \rangle$, and $\langle \cdot \rangle$ denotes the ensemble average over $\mathbf{u} \in \mathcal{S}$. If the design shape is defined such that the mean shape corresponds to the original design, then $\langle \boldsymbol{\delta}_s \rangle = 0$, $\forall \mathbf{x}$, and therefore $\hat{\boldsymbol{\delta}}_s = \boldsymbol{\delta}_s$.

The aim of the KLE is to find an optimal basis of orthonormal functions, for the linear representation of the deviation from the mean shape modification vector, expressed by

$$\hat{\boldsymbol{\delta}}_s(\mathbf{x},\mathbf{u}) = \sum_{k=1}^{\infty} \alpha_k(\mathbf{u}) \boldsymbol{\Phi}_k(\mathbf{x}) d\mathbf{x} \tag{2.23}$$

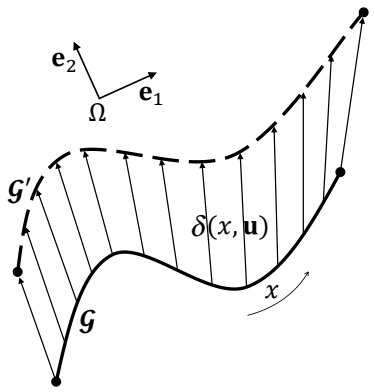


Figure 2.4: Scheme and notation for the current formulation, showing an example for $m = 1$ and $l = 2$.

where

$$\alpha_k(\mathbf{x}) = (\hat{\boldsymbol{\delta}}_s, \boldsymbol{\Phi}_k)_\rho = \int_{\mathcal{G}} \rho(\mathbf{x}) \hat{\boldsymbol{\delta}}_s(\mathbf{x}, \mathbf{u}) \cdot \boldsymbol{\Phi}_k(\mathbf{x}) d\mathbf{x} \tag{2.24}$$

are the basis-functions component, used hereafter as new design variables.

The optimality condition associated to the KLE refers to the geometric variance retained by the basis functions through Eq. 2.23. Combining Eqs. 2.22–2.24 yields

$$\sigma^2 = \sum_{k=1}^{\infty} \sum_{j=1}^{\infty} \langle \alpha_k \alpha_j \rangle (\boldsymbol{\Phi}_k, \boldsymbol{\Phi}_j)_\rho = \sum_{j=1}^{\infty} \langle \alpha_j^2 \rangle = \sum_{j=1}^{\infty} \left\langle (\hat{\boldsymbol{\delta}}, \boldsymbol{\Phi}_j)_\rho^2 \right\rangle \tag{2.25}$$

The basis retaining the maximum variance is so formed by those $\boldsymbol{\Phi}$ that are solutions of the variational problem

$$\begin{aligned} &\max_{\boldsymbol{\Phi} \in L^2_\rho(\mathcal{G})} J(\boldsymbol{\Phi}) = \left\langle (\hat{\boldsymbol{\delta}}_s, \boldsymbol{\Phi}_j)_\rho^2 \right\rangle \\ &\text{subject to } \ (\boldsymbol{\Phi}, \boldsymbol{\Phi})_\rho^2 = 1 \end{aligned} \tag{2.26}$$

which yields to

$$\mathcal{L}\boldsymbol{\Phi}(\mathbf{x}) = \int_{\mathcal{G}} \rho(\mathbf{y}) \left\langle \hat{\boldsymbol{\delta}}_s(\mathbf{x}, \mathbf{u}) \otimes \hat{\boldsymbol{\delta}}_s(\mathbf{y}, \mathbf{u}) \right\rangle \boldsymbol{\Phi}(\mathbf{y}) d\mathbf{y} = \lambda \boldsymbol{\Phi}(\mathbf{x}) \tag{2.27}$$

where $\mathcal{L}$ is the selfadjoint integral operator whose eigensolutions define the optimal basis functions for the linear representation of Eq. 2.23. Therefore, its eigenfunctions (KL modes) $\{\boldsymbol{\Phi}_k\}_{k=1}^{\infty}$ are orthogonal and form a complete basis for $L^2_\rho(\mathcal{G})$. Additionally, it may be proven [16] that

$$\sigma^2 = \sum_{j=1}^{\infty} \lambda_j \tag{2.28}$$

where the eigenvalues λ_j (KL values) represent the variance retained by the associated basis function $\boldsymbol{\Phi}_j$, through its component α_j in Eq. 2.23.

Finally, the solutions $\{\boldsymbol{\Phi}_k\}_{k=1}^{\infty}$ of Eq. 2.27 are used to build a reduced-dimensionality space for the shape modification. Assume that n, with $0 < n \leq 1$, is the desired level of confidence for the shape modification variability, Eq. 2.23 may be truncated to the N-th order, provided that

$$\sum_{k=1}^{N} \lambda_k \geq n \sum_{k=1}^{\infty} \lambda_k = n\sigma^2 \tag{2.29}$$

with $\lambda_k \geq \lambda_{k+1}$. Details of equations and numerical implementations are given in Ref. [16].

Chapter 3

Hydrodynamic solvers

"Contrariwise," continued Tweedledee, "if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic."

— Lewis Carroll, *Alice in Wonderland*

The prediction of ship hydrodynamics performance can be broken down into three general areas: (1) resistance and propulsion, (2) seakeeping, and (3) manoeuvring. There are several basic approaches to predict the hydrodynamics performance of a vessel, and they can be classified as: empirical/statistical, experimental (either in model- or full-scale), and numerical (either rather analytical or using computational fluid dynamics (CFD)).

In the simulation-based design optimization context, the "simulation" role is performed by the numerical approaches, specifically by low- and high-fidelity CFD solvers.

For ship resistance, CFD has become increasingly important and is now an indispensable part of the design process. Typically inviscid free-surface methods based on the boundary element approach are used to analyse the forebody, especially the interaction of bulbous bow and forward shoulder. Viscous flow codes focus on the aftbody or appendages. Flow codes modeling both viscosity and the wave-making are widely applied for flows involving breaking waves. So far CFD has been used to gain insight into local flow details and derive recommendation on how to improve a given design or select a most promising candidate design for model testing [98].

For seakeeping, simple strip methods are used to analyze the seakeeping properties. These usually employ boundary element methods to solve a succession of two-dimensional problems and integrate the results into a quasi-three-dimensional result with usually good accuracy.

Although a model of the final ship design is still tested in a towing tank, the testing sequence and content have changed significantly over time. Traditionally, unless the new ship design was close to an experimental series or a known parent ship, the design process incorporated many model tests. The process has been one of design, test, redesign, test, etc., sometimes involving more than ten models, each with slight variations. This is no longer feasible due to time-to-market requirements from shipowners and no longer necessary thanks to CFD developments. Combining CAD (computer-aided design) to generate new hull shapes in concert with CFD to analyze these hull shapes allows for rapid design explorations without model testing. With massive parallel computing and progress in optimization strategies, formal optimization of hulls, propellers, and appendages has drifted into industrial applications. CFD is increasingly used for the actual design of hull and propellers. Then often only the final design is actually tested to validate the intended performance features and to get a power prediction accepted in practice as highly accurate. As a consequence of this practice, model tests for shipyard customers have declined considerably since the 1980s. This was partially compensated for by more sophisticated and detailed tests funded from research projects to validate and calibrate CFD methods.

3.1 Fluid mechanics equations

For the velocities involved in ship flows, water can be regarded as incompressible, i.e. the density ρ is constant. Therefore we will limit ourselves here to incompressible flows. All equations are given in a Cartesian coordinate system with z pointing downwards. The continuity equation states that any amount flowing into a control volume also flows out of the control volume at the same time. Consider an infinitely small control volume (see Fig. 3.1) for the two-dimensional case, where u and v are the velocity components in x and y direction, respectively. The indices denote partial derivatives, e.g. $u_x = \partial u/\partial x$. Positive mass flux leaves the control volume; negative mass

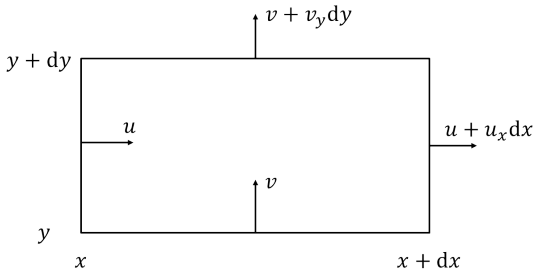


Figure 3.1: Control volume to derive continuity equation in two dimensions

flux enters the control volume. The total mass flux has to fulfill:

$$-\rho dy\, u + \rho dy(u + u_x dx) - \rho dx\, v + \rho dx(v + v_y dy) = 0 \tag{3.1}$$

$$u_x + v_y = 0 \tag{3.2}$$

The continuity equation in three dimensions can be derived correspondingly to:

$$u_x + v_y + w_z = 0 \tag{3.3}$$

where w is the velocity component in z direction.

The Navier–Stokes equations together with the continuity equation suffice to describe all real flow physics for ship flows. The Navier–Stokes equations describe conservation of momentum in the flow:

$$\begin{aligned} \rho(u_t + uu_x + vu_y + wu_z) &= \rho f_1 - p_x + \mu(u_{xx} + u_{yy} + u_{zz}) \\ \rho(v_t + uv_x + vv_y + wv_z) &= \rho f_2 - p_y + \mu(v_{xx} + v_{yy} + v_{zz}) \\ \rho(w_t + uw_x + vw_y + ww_z) &= \rho f_3 - p_z + \mu(w_{xx} + w_{yy} + w_{zz}) \end{aligned} \tag{3.4}$$

where f_i is an acceleration due to a volumetric force, p the pressure, μ the viscosity, and t the time. Often the volumetric forces are neglected, but gravity can be included by setting f_3 =g. The left hand side of the Navier–Stokes equations without the time derivative describes convection, the time derivative describes the rate of change (source term), the last term on the right hand side describes diffusion. The Navier–Stokes equations in the above form contain on the light hand side products of the velocities and their derivatives. This is a non-conservative formulation of the Navier–Stokes equations. A conservative formulation contains unknown functions (here velocities) only as first derivatives. Using the product rule for differentiation and the continuity equation (Eq. 3.3), the non-conservative formulation can be transformed into a conservative formulation, e.g. for the first of the Navier–Stokes equations above:

$$\begin{aligned} (u^2)_x + (uv)_y + (uw)_z &= 2uu_x + u_y v + uv_y + u_z w + uw_z \\ &= uu_x + vu_y + wu_z + u(u_x + v_y + w_z) \\ &= uu_x + vu_y + wu_z \end{aligned} \tag{3.5}$$

Navier–Stokes equations and the continuity equation form a system of coupled, nonlinear partial differential equations. An analytical solution of this system is impossible for ship flows. Even if the influence of the free surface (waves) is neglected, todays computers are not powerful enough to allow a numerical solution either. Even if such a solution may become feasible in the future, it is questionable if it is really necessary for engineering purposes in naval architecture.

Velocities and pressure may be divided into a time average and a fluctuation part to bring the Navier–Stokes equations closer to a form where a numerical solution is possible. Time averaging yields the Reynolds-averaged Navier–Stokes (RANS) equations, therefore u, v, w, and p are from now on time averages, whereas u', v', and w' denote the fluctuation parts. For unsteady flows, high-frequency fluctuations are averaged over a chosen time interval (assembly average). This time interval is small compared to the global motions, but large compared to the turbulent fluctuations. Most computations for ship flows are limited to steady flows where the terms u_t, v_t, and w_t vanish.

The RANS equations have a similar form to the Navier–Stokes equations:

$$\begin{aligned} \rho(u_t + uu_x + vu_y + wu_z) &= \rho f_1 - p_x + \mu(u_{xx} + u_{yy} + u_{zz}) - \rho((\overline{u'u'})_x + (\overline{u'v'})_y + (\overline{u'w'})_z) \\ \rho(v_t + uv_x + vv_y + wv_z) &= \rho f_2 - p_y + \mu(v_{xx} + v_{yy} + v_{zz}) - \rho((\overline{u'v'})_x + (\overline{v'v'})_y + (\overline{v'w'})_z) \\ \rho(w_t + uw_x + vw_y + ww_z) &= \rho f_3 - p_z + \mu(w_{xx} + w_{yy} + w_{zz}) - \rho((\overline{u'w'})_x + (\overline{v'w'})_y + (\overline{w'w'})_z) \end{aligned} \tag{3.6}$$

including the derivatives of the Reynolds stresses (last term on the right hand side of Eq. 3.6). The time averaging eliminated the turbulent fluctuations in all terms except the Reynolds stresses. The RANS equations require a turbulence model that couples the Reynolds stresses to the average velocities.

Neglecting viscosity and all turbulence effects turns the RANS equations into the Euler equations, which still have to be solved together with the continuity equations:

$$\begin{aligned} \rho(u_t + uu_x + vu_y + wu_z) &= \rho f_1 - p_x \\ \rho(v_t + uv_x + vv_y + wv_z) &= \rho f_2 - p_y \\ \rho(w_t + uw_x + vw_y + ww_z) &= \rho f_3 - p_z \end{aligned} \tag{3.7}$$

Euler solvers allow coarser grids and are numerically more robust than RANS solvers. They are suited for computation of flows about lifting surfaces (foils) and are thus popular in aerospace applications. They are not so well suited for ship flows and generally not recommended because they combine the disadvantages of RANS and Laplace solvers without being able to realize their major advantages: programming is almost as complicated as for RANS solvers, but the physical model offers hardly any improvements over simple potential flow codes (Laplace solvers) [98].

A further simplification is the assumption of irrotational flow:

$$\nabla \times \mathbf{v} = 0 \tag{3.8}$$

A flow that is irrotational, inviscid and incompressible is called potential flow. In potential flows the components of the velocity vector are no longer independent from each other. They are coupled by the potential ϕ. The derivative of the potential in arbitrary direction gives the velocity component in this direction:

$$\mathbf{v} = \nabla\phi \tag{3.9}$$

Three unknowns (the velocity components) are thus reduced to one unknown (the potential). This leads to a considerable simplification of the computation.

The continuity equation simplifies to Laplaces equation for potential flow:

$$\Delta\phi = \phi_{xx} + \phi_{yy} + \phi_{zz} = 0 \tag{3.10}$$

If the volumetric forces are limited to gravity forces, the Euler equations can be written as:

$$\nabla\left(\phi_t + \frac{1}{2}(\nabla\phi)^2 - gz + \frac{p}{\rho}\right) = 0 \tag{3.11}$$

Integration gives Bernoullis equation:

$$\phi_t + \frac{1}{2}(\nabla\phi)^2 - gz + \frac{p}{\rho} = \text{const.} \tag{3.12}$$

The Laplace equation is sufficient to solve for the unknown velocities. The Laplace equation is linear. This offers the big advantage of combining elementary solutions (so-called sources, sinks, dipoles, vortices) to arbitrarily complex solutions. Potential flow (PF) codes are still largely used tools in ship and propeller design.

3.2 Hydrodynamic tools

In the present work, the simulation-based design optimization problems, used as benchmark to test and assess the performance of the optimization algorithm, pertain the hull-form optimization for calm-water resistance reduction and improved seakeeping performance of two different kind of vessels. The analysis tools used both low- and high-fidelity solvers, and they are briefly introduced in the following subsections.

3.2.1 WAve Resistance Program

The WAve Resistance Program (WARP) is a code developed at CNR-INSEAN. The mathematical model is approximated by a Boundary Element Method, in which the integral equation on the boundary is approximated by a piecewise constant distribution of source density on quadrilateral plane panels. Wave resistance computations are based on the linear PF theory [99]. The simplest linear formulation (Kelvin linearization) is obtained by assuming that the actual flow is slightly perturbed from the free stream, and its potential function is given by $\phi = Ux + \tilde{\varphi}$, which provides the Neumann-Kelvin (NK) problem for the Laplace equation. A further linearization, suggested

by Dawson [100], is based on the assumption that the flow near the body is perturbed around the double model (DM) flow, and its potential function is given by $\phi = Ux + \varphi_d + \tilde{\varphi}$. NK is usually reasonable for slender bodies and high speeds, whereas DM is usually more suitable for wider bodies and low speeds. The wave resistance can be evaluated by both a pressure integral over the body surface and the transverse wave cut method [101], whereas the frictional resistance is estimated using a flat-plate approximation, based on the local Reynolds number [102]. The steady 2DOF (sinkage and trim) equilibrium is achieved by iteration of the flow solver and the body equation of motion.

3.2.2 Ship Motion Program

The Standard Ship Motion program (SMP) was developed at the David Taylor Naval Ship Research and Development Center in 1981, as a prediction tool for use in the Navys ship design process. SMP is a potential flow code based on linearized strip theory. It provides a predictions of the motions, i.e., displacements, velocities, and accelerations, for a ship advancing at constant speed, with arbitrary heading in both regular waves and irregular seas. The irregular seas are modelled using a two-parameter Bretschneider wave spectral model. Both long-crested and short-crested results are provided. In addition to the 6DOF responses, the absolute motion, velocity, acceleration, as well as the relative motion and velocity for various locations on the ship can also be obtained. The probabilities and frequencies of submergence, emergence, and/or slamming occurrence for various locations on the ship, are also available. Vertical shears and bending moments due to the ship motions can be produced [103].

3.2.3 CFDShip-Iowa

CFDShip-Iowa is a general-purpose unsteady Reynolds-averaged Navier-Stokes (URANS) CFD code that has been developed at the University of Iowa, IIHR-Hydroscience & Engineering, over the past 25 years, to handle a broad range of ship hydrodynamics problems. Originally designed to support both thesis and project research in the areas of resistance and propulsion, it has been successfully transitioned to Navy and university laboratories and industry, and has recently been extended to unsteady applications such as seakeeping and maneuvering. It was developed following a modern software-development philosophy, which was based upon open source, revision control, modular coding using Fortran 90/95, liberal use of comments, and an easy to understand architecture that enables model development by users.

The equations are solved in an inertial coordinate system, either fixed to a ship or other frame moving at constant speed or in the earth system. The free surface is modelled with a single-phase capturing approach, which means that only the water flow is solved, enforcing kinematic and dynamic free surface boundary conditions on the interface. Arbitrary free surface topologies can be predicted, with the limitation that pressurized closed air/water packets (bubbles) cannot be simulated. It uses structured multiblock grids, and has overset capabilities. Capabilities include 6DOF motions, several turbulence models, moving control surface, multi-objects, advanced controllers, propulsion models, incoming waves and wind, bubbly flow, and fluid-structure interaction.

Details of equations, numerical implementations and validation of the numerical solver are given in [104].

Chapter 4

Optimization problems

"The time has come," the walrus said,"to talk of many things: Of shoes and ships - and sealing wax - of cabbages and kings"

— Lewis Carroll, *Alice in Wonderland*

4.1 Analytical test functions

A benchmark of seventy-three well-known analytical test functions [24, 44, 65, 105, 106, 107], including simple unimodal (i.e., those denoted by a superscript "*" in Tab. 4.1), highly complex multimodal, and not differentiable problems, has been used in order to assess the algorithms performance. Table 4.1 summarizes the test functions used herein, and the algorithm applied on each one. The analytical expressions of the test functions are reported in Appendix A.

In the following, DPSO, LS-DF_PSO, DIRMIN, DIRMIN-2, and FSA setting parameters are presented and discussed. The different setups have been used to verify and assess the algorithm performances. The results are presented in the next chapter.

4.1.1 DPSO setting parameters

The effectiveness and efficiency of DPSO for box constrained optimization are significantly influenced by four main setting parameters: (a) the number of swarm particles interacting during the optimization, (b) the initialization of the particles in terms of initial location and velocity, (c) the set of coefficients defining the personal or social behavior of the swarm dynamics, and (d) the method to handle the box constraints. These parameters and their effects on PSO have been studied by a number of authors [33, 50, 59] and a preliminary assessment of the performances of DPSO, varying (a), (b) and (c), is presented in Ref. [44].

In order to identify of the most effective and efficient parameters for both synchronous and asynchronous deterministic particle swarm optimization (S-DPSO and A-DPSO), for their use in SBD procedures, the DPSO parameters are defined in the following, and applied on 60 test functions (see Tab. 4.1), with dimensionality ranging from two to twenty. Their full-factorial combination is considered, resulting in a total of 420 DPSO setups. The simulation budget (maximum number of objective function evaluations) is studied up to 4096 times the number of variables.

Number of particles

The number of particles used (N_p) is defined as

$$N_p = 2^m N, \quad \text{with} \quad m \in \mathbb{N}[1,7] \tag{4.1}$$

therefore ranging from $2N$ to $128N$, where N is the number of design variables.

Table 4.1: Analytical test functions

f_p	Name	N	DPSO	LS-DF_PSO	DIRECT	DIRMIN	DIRMIN-2	FSA
f_1	Ackley	2	✓	✓	✓	✓	✓	✓
f_2	Ackley	10	–	–	✓	✓	✓	–
f_3	Ackley	30	–	–	✓	✓	✓	–
f_4	Ackley	50	–	–	✓	✓	✓	–
f_5	Alpine	2	✓	✓	✓	✓	✓	✓
f_6	Alpine	5	✓	✓	✓	✓	✓	✓
f_7	Alpine	10	✓	✓	✓	✓	✓	✓
f_8	Alpine	20	✓	✓	✓	✓	✓	✓
f_9	Beale	2	✓	✓	✓	✓	✓	✓
f_{10}	Booth*	2	✓	✓	✓	✓	✓	✓
f_{11}	Bukin n.6	2	✓	✓	✓	✓	✓	✓
f_{12}	Colville	4	✓	✓	✓	✓	✓	✓
f_{13}	Cosine Mixture	2	✓	✓	✓	✓	✓	✓
f_{14}	Cosine Mixture	4	✓	✓	✓	✓	✓	✓
f_{15}	Dixon-Price*	2	✓	✓	✓	✓	✓	✓
f_{16}	Dixon-Price*	5	✓	✓	✓	✓	✓	✓
f_{17}	Dixon-Price*	10	✓	✓	✓	✓	✓	✓
f_{18}	Dixon-Price*	20	✓	✓	✓	–	–	✓
f_{19}	Dixon-Price*	25	–	–	✓	✓	✓	–
f_{20}	Dixon-Price*	50	–	–	✓	✓	✓	–
f_{21}	Easom	2	✓	✓	✓	✓	✓	✓
f_{22}	Exponential	2	✓	✓	✓	✓	✓	✓
f_{23}	Exponential	4	✓	✓	✓	✓	✓	✓
f_{24}	Freudenstein-Roth	2	✓	✓	✓	✓	✓	✓
f_{25}	Goldstein-Price	2	✓	✓	✓	✓	✓	✓
f_{26}	Griewank	2	✓	✓	✓	✓	✓	✓
f_{27}	Griewank	5	✓	✓	✓	✓	✓	✓
f_{28}	Griewank	10	✓	✓	✓	✓	✓	✓
f_{29}	Griewank	20	✓	✓	✓	✓	✓	✓
f_{30}	Hartman n.3	3	✓	✓	✓	✓	✓	✓
f_{31}	Hartman n.6	6	✓	✓	✓	✓	✓	✓
f_{32}	Levy 5^n	2	✓	✓	✓	✓	✓	✓
f_{33}	Levy 5^n	5	✓	✓	✓	✓	✓	✓
f_{34}	Levy 5^n	10	✓	✓	✓	✓	✓	✓
f_{35}	Levy 5^n	20	✓	✓	✓	✓	✓	✓
f_{36}	Levy 10^n	2	✓	✓	✓	✓	✓	✓
f_{37}	Levy 10^n	5	✓	✓	✓	✓	✓	✓
f_{38}	Levy 10^n	10	✓	✓	✓	✓	✓	✓
f_{39}	Levy 10^n	20	✓	✓	✓	✓	✓	✓
f_{40}	Levy 15^n	2	✓	✓	✓	✓	✓	✓
f_{41}	Levy 15^n	5	✓	✓	✓	✓	✓	✓
f_{42}	Levy 15^n	10	✓	✓	✓	✓	✓	✓
f_{43}	Levy 15^n	20	✓	✓	✓	✓	✓	✓
f_{44}	Matyas*	2	✓	✓	✓	✓	✓	✓
f_{45}	Multi Modal	2	✓	✓	✓	✓	✓	✓
f_{46}	Multi Modal	5	✓	✓	✓	✓	✓	✓
f_{47}	Multi Modal	10	✓	✓	✓	✓	✓	✓
f_{48}	Multi Modal	20	✓	✓	✓	✓	✓	✓
f_{49}	Powell*	8	–	–	✓	✓	✓	–
f_{50}	Powell*	16	–	–	✓	✓	✓	–
f_{51}	Powell*	24	–	–	✓	✓	✓	–
f_{52}	Quartic*	2	✓	✓	✓	✓	✓	✓
f_{53}	Rastrigin	10	–	–	✓	✓	✓	–
f_{54}	Rastrigin	30	–	–	✓	✓	✓	–
f_{55}	Rastrigin	50	–	–	✓	✓	✓	–
f_{56}	Rosenbrock*	2	✓	✓	✓	✓	✓	✓
f_{57}	Schaffer n.2	2	✓	✓	✓	✓	✓	✓
f_{58}	Schaffer n.6	2	✓	✓	✓	✓	✓	✓
f_{59}	Schwefel	10	–	–	✓	✓	✓	–
f_{60}	Schwefel	20	–	–	✓	✓	✓	–
f_{61}	Shekel n.5	4	✓	✓	✓	✓	✓	✓
f_{62}	Shekel n.7	4	✓	✓	✓	✓	✓	✓
f_{63}	Shekel n.10	4	✓	✓	✓	✓	✓	✓
f_{64}	Shubert penalty 1	2	✓	✓	✓	✓	✓	✓
f_{65}	Shubert penalty 2	2	✓	✓	✓	✓	✓	✓
f_{66}	Six-Hump Camel Back	2	✓	✓	✓	✓	✓	✓
f_{67}	Sphere*	2	✓	✓	✓	✓	✓	✓
f_{68}	Styblinski-Tang	2	✓	✓	✓	✓	✓	✓
f_{69}	Styblinski-Tang	4	✓	✓	✓	✓	✓	✓
f_{70}	Test Tube Holder	2	✓	✓	✓	✓	✓	✓
f_{71}	Three-Hump Camel Back	2	✓	✓	✓	✓	✓	✓
f_{72}	Treccani*	2	✓	✓	✓	✓	✓	✓
f_{73}	Tripod	2	✓	✓	✓	✓	✓	✓
	Total		60	60	73	72	72	60

* Denotes unimodal functions

Particles initialization

The initialization of particles' location and velocity is performed using a deterministic and homogeneous distribution, following the Hammersley sequence sampling (HSS) [108]. Specifically, let $\mathbf{p} = \{p_1, \ldots, p_{N-1}\}$ be a vector of prime numbers with $p_j \neq p_i$, $\forall j \neq i$. Any positive integer i can be expressed using the sequence $\{p_i\}$ by

$$j = \sum_{k=0}^{r} a_k p_i^k \tag{4.2}$$

where r is a suitable integer and a_k is an integer in $[0, p_i - 1]$. Finally, the j-th particle location is defined as

$$\mathbf{x}_j = \left\{ \frac{j}{N_p}, \phi_{p_1}(j), \cdots, \phi_{p_{N-1}}(j) \right\} \quad \text{for } j = 0, 1, 2, ..., N_p - 1 \tag{4.3}$$

where $\phi_{p_i}(j) = \sum_{k=0}^{r} a_k / p_i^{k+1}$. The Eq. 4.3 is applied to three different sub-domains, defined as:

A. entire domain $\mathcal{D}$ (red dots in Fig. 4.1a)

B. domain bounds (blu triangles in Fig. 4.1b)

C. domain and bounds (red dots and blue triangles in Fig. 4.1c)

On the other hand, the initial velocity is defined by either the following:

- null velocity:

$$\mathbf{v}_j^0 = 0, \qquad \forall\, j \in [1, N_p] \tag{4.4}$$

- non-null velocity, based on initial particle position:

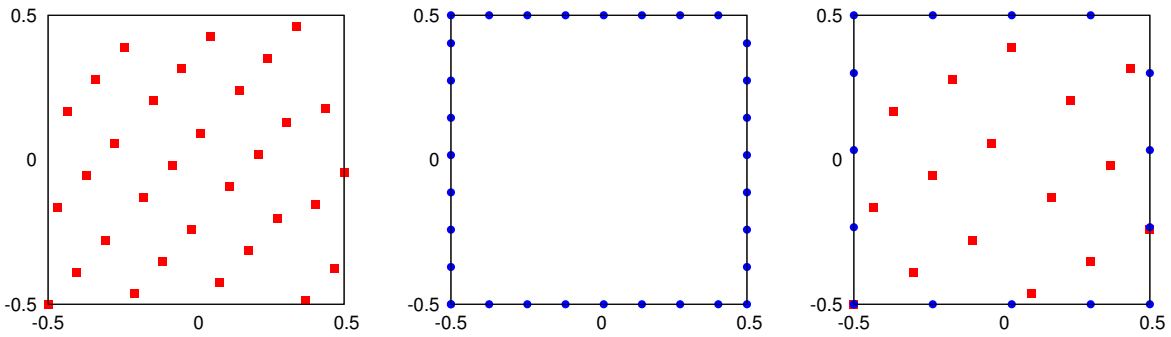
$$\mathbf{v}_j^0 = \frac{2}{\sqrt{N}} \left(\mathbf{x}_j^0 - \frac{\mathbf{l} + \mathbf{u}}{2} \right) \tag{4.5}$$

  where $\mathbf{l}$ and $\mathbf{u}$ represent the lower and upper bound for $\mathbf{x}$, respectively [47].

Combining initial position and velocity settings results in six different initializations, summarized in Tab. 4.5.

Table 4.2: DPSO, particles initialization

HSS, sub-domain	$\mathbf{v} = 0$	$\mathbf{v} \neq 0$
Domain	A.0	A.1
Bounds	B.0	B.1
Domain and bounds	C.0	C.1



(a) Domain (b) Bounds (c) Domain and bounds

Figure 4.1: Examples of initializations in $\mathcal{D} = [-0.5, 0.5] \times [-0.5, 0.5]$ with 32 individuals

Coefficients set

Five coefficients sets are taken from literature, as proposed by several authors. The first set is the original by Shi and Eberhart [50], the second was suggested by Carlisle and Dozier [109], the third was proposed by Trelea [59], the fourth is a further suggestion by Clerc [110], whereas the fifth was suggested by Peri and Tinti [111]. The associated numerical values are included in Tab. 4.3, and they all satisfy Eq. 1.8.

Table 4.3: DPSO, coefficient set

Set ID	Name	χ	c_1	c_2	β
1	Shi and Eberhart (1998)	0.729	2.050	2.050	0.864
2	Carlisle and Dozier (2001)	0.729	2.300	1.800	0.864
3	Trelea (2003)	0.600	1.700	1.700	0.638
4	Clerc (2006)	0.721	1.655	1.655	0.693
5	Peri and Tinti (2012)	0.754	2.837	1.597	0.953

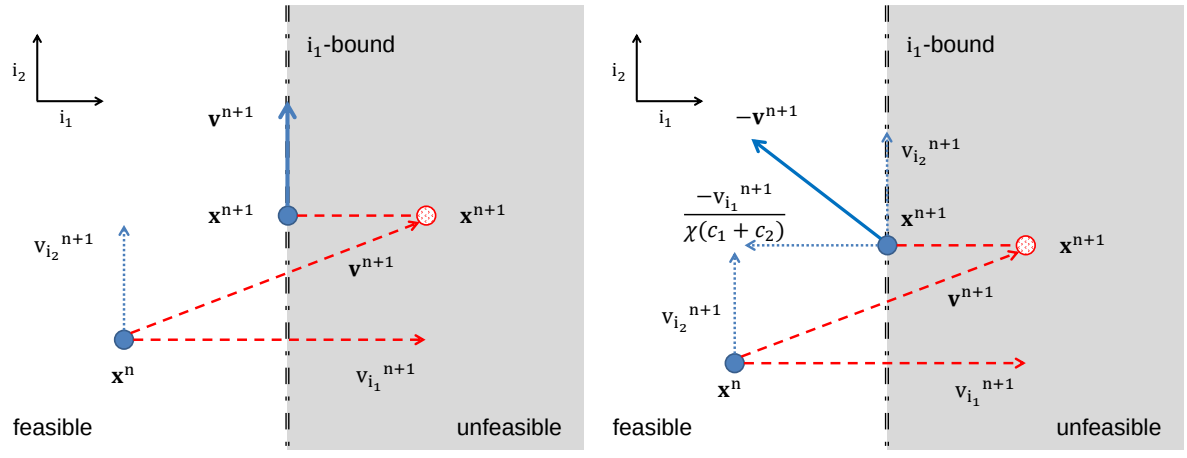
Box constraints

The original PSO [30] provides an update of position and velocity of particles for unconstrained problems. Thus, in case possible constraints are present, the original formulation might be inadequate, even in case simple box constraints on the unknowns are considered. This implies that, during the evolution of the swarm, simply using Eq. 1.5 the particles can travel outside the domain bounds. This can be a critical issue in SBD problems, when the domain bounds cannot be violated due to physical/geometrical/grids constraints. A barrier (wall) or a penalty approach can be used on the bounds of the research space, in order to confine the particles [112, 113]. Herein, the approach presented in [114] is applied.

Specifically, the particles are confined within $\mathcal{D}$ using an inelastic wall-type approach (IW). If the j-th particle is found to violate one of the bounds in the transition from n-th to $(n+1)$-th DPSO iteration, it is placed on the bound setting to zero the associated velocity component (see Fig.4.2a). This approach helps the algorithm to explore the domain bounds. The IW approach is implemented in Alg. 8.

Algorithm 8 DPSO, Inelastic wall-type approach

```
for j = 1, number of particles do
    for i = 1, number of variables do
        if x^n_{j,i} > u_i then
            x^n_{j,i} = u_i,   v^n_{j,i} = 0
        else {x^n_{j,i} < l_i}
            x^n_{j,i} = l_i,   v^n_{j,i} = 0
        end if
    end for
end for
```



(a) IW (b) SEW

Figure 4.2: Wall-type approaches applied to the j-th particle in the transition from n-th to $(n+1)$-th DPSO iteration

The use of IW has some limitations: in the unlikely event that all the particles tend to leave the domain from the same hyper-corner, the IW sets all velocities to zero and the DPSO progress may stop prematurely. For this reason, a semi-elastic wall-type approach (SEW) is also used in this work. Specifically, in case the particle position violates a bound constraint, then the particle position is modified in order to make that constraint active (i.e. the particle is moved on the boundary of that constraint), while the associated velocity component is defined as follows (see also Fig. 4.2b):

Algorithm 9 DPSO, Semi-elastic wall-type approach

for $j = 1$, number of particles **do**
  for $i = 1$, number of variables **do**
    if $x^n_{j,i} > u_i$ **then**
      $v^n_{j,i} = -\frac{v^n_{j,i}}{\chi(c_1+c_2)}$
    else $\{x^n_{j,i} < l_i\}$
      $v^n_{j,i} = -\frac{v^n_{j,i}}{\chi(c_1+c_2)}$
    end if
  end for
end for

Observe that the damping factor $[\chi(c_1+c_2)]^{-1}$ in Alg. 9 is used to confine the particle in the feasible domain.

Number of function evaluations and DPSO iterations

The number of function evaluations $N_{\max}$ (evaluations budget) is defined as

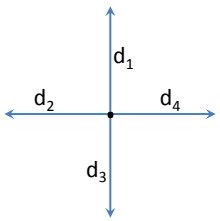
$$N_{\max} = 2^c N, \quad \text{where} \quad c \in \mathbb{N}[7,12] \tag{4.6}$$

and therefore ranges from $128N$ to $4096N$. Similarly to the setting in Eq. 4.9, the number of DPSO iterations N_{iter} is set as

$$N_{iter} = \frac{N_{\max}}{N_p} = \frac{2^c N}{2^m N} = 2^{c-m} \tag{4.7}$$

4.1.2 LS-DF_PSO setting parameters

Global convergence properties of a modified DPSO scheme may be obtained by properly combining DPSO with a line search-based derivative-free method, so that convergence to stationary points can be forced at a reasonable cost. Ref. [23] provides a robust method to force the convergence of a subsequence of points toward a stationary point, which satisfies first order optimality conditions for the objective function. The method, namely LS-DF_PSO, starts by coupling the DPSO scheme with a line search-based method. Specifically, a Positively Spanning Set (PSS) is used, where the set of search directions ($D_\oplus$) is defined by the unit vectors $\pm e_i$, $i = 1,\dots,N$, as shown in the following equation (i.e., $N = 2$) and in Fig. 4.3.



$$D_\oplus = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} \tag{4.8}$$

Figure 4.3: Example of PSS in $\mathbb{R}^2$

After each DPSO iteration, the local search is performed, from the best particle, if the swarm has not find a new global minimum. The initial step size (ς^k) for the local search is set equal to 0.25 times the variable domain range, and it is reduced by $\vartheta = 0.5$ at each local search iteration. Local searches continue in each direction until the step size is greater than $\mu = 10^{-3}$. If the local search stops without providing a new global minimum, the actual global minimum is declared as a stationary point. The line search method is not allowed to violate the box constraints.

The method is applied on 60 analytical test function (see Tab. 4.1) with dimensionality ranging from two to twenty, and a limit to the maximum number of function evaluations is set equal to 800 times the number of variables.

4.1.3 DIRMIN and DIRMIN-2 setting parameters

Seventy-two analytical test functions, with N ranging from 2 to 50, have been used to assess the algorithms performances. A limit on the maximum number of function evaluations $N_{\max}$ is imposed equal to $256N$. The derivative-free local minimizations in DIRMIN and DIRMIN-2 start when the number of function evaluations reaches the activation trigger $\gamma N_{\max}$, where $\gamma \in [0,1]$. The local minimizations proceed until either the number of function evaluations exceeds $N_{\max}$ or the stepsize falls below the given tolerance $\beta > 0$. The values that can be assumed by the parameters γ and β are summarized in Tab. 4.4. The algorithms with the respective parameter value combinations are numbered from 1 to 25.

Table 4.4: DIRMIN and DIRMIN-2 setup ID

$\beta \backslash \gamma$	0.00	0.25	0.50	0.75	1.00
1e-5	1	2	3	4	5
1e-4	6	7	8	9	10
1e-3	11	12	13	14	15
1e-2	16	17	18	19	20
1e-1	21	22	23	24	25

4.1.4 FSA setting parameters

The effectiveness and efficiency of FSA are influenced, as DPSO, by the choice of four main parameters: (a) the number of individuals interacting during the optimization, (b) the initialization of the shoal in terms of initial location and speed, (c) the set of coefficients controlling the shoal dynamics, and (d) the method to handle the box constraints.

The approach for the analysis includes a parametric study using 60 analytical test functions (see Tab. 4.1) characterized by different degrees of nonlinearities and number of local minima, with full-factorial combination of: (a) number of individuals, using power of two per number of design variables, (b) initialization of the shoal, in terms of initial position and velocity, by Hammersley distributions [108], (c) twenty-seven different set of coefficients, and (d) the method to handle the box constraints.

The FSA parameters used in the current analysis are defined in the following subsection. Their full-factorial combination is considered, resulting in a total of 486 FSA setups. The simulation budget is studied up to 4096 times the number of variables.

Fish shoal size

The number of individuals used (N_s) is defined as

$$N_s = 2^m N, \quad \text{with} \quad m \in \mathbb{N}[2,4] \tag{4.9}$$

therefore ranging from $4N$ to $16N$.

Shoal initialization

The initialization of individuals' location and velocity is performed using a deterministic and homogeneous distribution, following the Hammersley sequence sampling (HSS) [108], as for DPSO (see Sec. 4.1.1).

The Eq. 4.3 is applied to three different unit sub-domains, defined as:

A. domain $\mathcal{U}$ (red squares in Fig. 4.1a)

B. domain bounds (blu dots in Fig. 4.1b)

C. domain and bounds (red squares and blue dots in Fig. 4.1c)

On the other hand a non-null initial velocity is used and defined by the following

$$\mathbf{v}_j^0 = \frac{2}{\sqrt{N}} \left(\mathbf{x}_j^0 - \frac{\mathbf{l}+\mathbf{u}}{2} \right) \tag{4.10}$$

where $\mathbf{l}$ and $\mathbf{u}$ represent the lower and upper bound for $\mathbf{x}$ respectively [47], resulting in three different initializations, summarized in Tab. 4.5.

Table 4.5: FSA, individuals initialization

HSS, sub-domain	ID
Domain	A
Bounds	B
Domain and bounds	C

Shoal dynamics controlling coefficients

Provided that all the variables are normalized such that the research space $\mathcal{D}$ is confined in a unit hyper-cube $\mathcal{U}$ (i.e. $-0.5 \leq \mathbf{x} \leq 0.5$), the following positions are used for the coefficients controlling the shoal dynamics

$$\xi \Delta t < 1, \qquad k = h = \frac{q}{N_s}, \qquad \Delta t = \frac{\Delta t_{\max}}{p} \tag{4.11}$$

where $q \in \mathbb{R}^+$ is the weight for the shoal attraction force ($\boldsymbol{\delta}_j$) and the food-related attraction force ($\boldsymbol{\varphi}_j$), and $p \in \mathbb{R}^+$ define the time step of the integration scheme. This two parameters within the damping factor ξ correspond to the FSA coefficients set. Table 4.6 summarizes the coefficient sets used herein.

Table 4.6: FSA, coefficients set

ID	ξ	q	p
1	0.01	0.10	2.00
2	0.10	1.00	4.00
3	1.00	10.0	8.00

Box constraints

The FSA formulation proposed herein provides an update of position and velocity of individuals for unconstrained problems. This implies that, during the evolution of the shoal, the individuals can travel outside the domain bounds. This can be a critical issue in SBD problems, when the domain bounds cannot be violated due to physical/geometrical/grids constraints. A barrier (wall) or a penalty approach can be used on the bounds of the research space, in order to confine the individuals. Herein, the approach presented for DPSO (see Sec. 4.1.1) is applied.

Specifically, the individuals are confined within $\mathcal{U}$ using an inelastic wall-type approach (IW) as in Alg. 10. If an individual is found to violate one of the bounds in the transition from n-th to (n+1)-th FSA iteration, it is placed on the bound setting to zero the associated velocity component (see Fig. 4.4a). This approach helps the algorithm to explore the domain bounds.

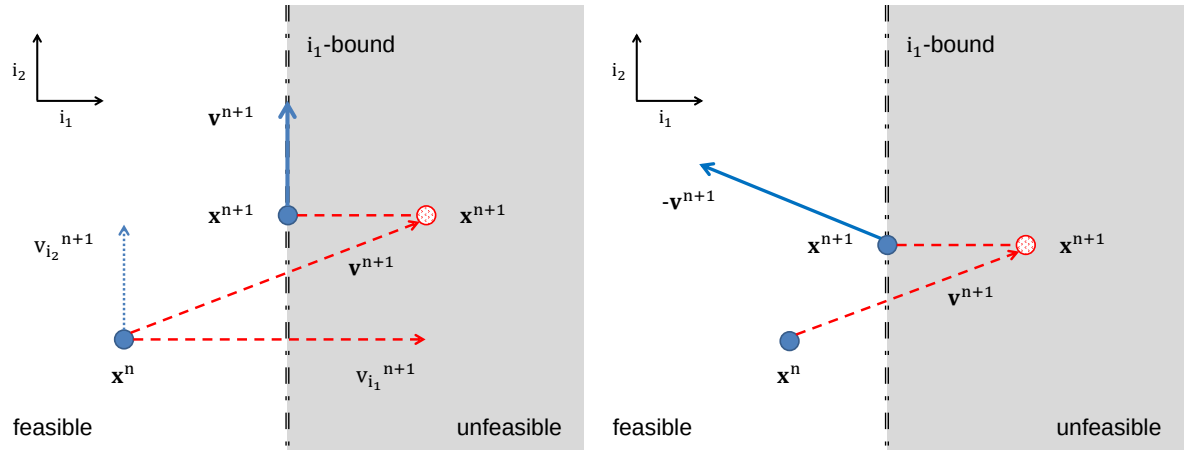
Algorithm 10 FSA, Inelastic wall-type approach

```
for j = 1, number of individuals do
  for i = 1, number of variables do
    if x^n_{j,i} > u_i then
      x^n_{j,i} = u_i,   v^n_{j,i} = 0
    else {x^n_{j,i} < l_i}
      x^n_{j,i} = l_i,   v^n_{j,i} = 0
    end if
  end for
end for
```

The use of IW has some limitations: in the unlikely event that all the particles tend to leave the domain from the same hyper-corner, the IW sets all velocities to zero and the FSA progress may stop prematurely. For this reason, an elastic wall-type approach (EW) is also used in this work. Specifically, in case the individual position violates a bound constraint, then the individual position is modified in order to make that constraint active (i.e. the individual is moved on the boundary of that constraint), while the associated velocity component is defined as follows (see also Fig. 4.4b):

Algorithm 11 FSA, Elastic wall-type approach

```
for j = 1, number of individuals do
  for i = 1, number of variables do
    if x^n_{j,i} > u_i then
      x^n_{j,i} = u_i,   v^n_{j,i} = -v^n_{j,i}
    else {x^n_{j,i} < l_i}
      x^n_{j,i} = l_i,   v^n_{j,i} = -v^n_{j,i}
    end if
  end for
end for
```



(a) Inelastic wall-type approach (b) Elastic wall-type approach

Figure 4.4: Wall-type approaches applied for the j-th individual in the transition from n-th to $(n+1)$-th FSA iteration

Number of function evaluations and FSA iterations

The number of function evaluations $N_{\max}$ (evaluations budget) is defined as

$$N_{\max} = 2^c N, \quad \text{where} \quad c \in \mathbb{N}[7,12] \tag{4.12}$$

and therefore ranges from $128N$ to $4096N$. As per Eq. 4.9, the number of FSA iterations N_{iter} is

$$N_{iter} = \frac{N_{\max}}{N_s} = \frac{2^c N}{2^m N} = 2^{c-m} \tag{4.13}$$

4.1.5 Evaluation metrics

The metrics used to assess the algorithms performance and identify the most promising setup for DPSO, DIRMIN, DIRMIN-2, and FSA are presented in the following.

Absolute metrics

Three absolute performance criteria are used to assess the algorithms' performances and defined as follows [44]:

$$\Delta_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_{i,\min} - x^\star_{i,\min}}{R_i}\right)^2}, \qquad \Delta_f = \frac{f_{\min} - f^\star_{\min}}{f^\star_{\max} - f^\star_{\min}}, \qquad \Delta_t = \sqrt{\frac{\Delta_x^2 + \Delta_f^2}{2}} \tag{4.14}$$

Δ_x is a normalized Euclidean distance between the minimum position found by the algorithm ($\mathbf{x}_{\min}$) and the analytical minimum position ($\mathbf{x}^\star_{\min}$), where $R_i = |u_i - l_i|$ is the range of the i-th design variable. Δ_f is the associated normalized distance in the image space, $f_{\min}$ is the minimum found by the algorithm, $f^\star_{\min}$ is the analytical minimum, and $f^\star_{\max}$ is the analytical maximum of the function $f(\mathbf{x})$ in the search domain $\mathcal{D}$. Δ_t is a combination of Δ_x and Δ_f and used for an overall assessment.

Additionally, the relative variability $\sigma^2_{r,k}$ [24] for each metric Δ_x, Δ_f, Δ_t (Eq. 4.14) is used to assess the impact of each tuning parameter s_k on the algorithms' performance. Defining the algorithm's tuning parameter vector as $s = [s_1, s_2, \ldots, s_S]^T \in \mathbb{R}^S$, the relative performance variability associated to its k-th component is

$$\sigma^2_{r,k} = \frac{\sigma^2_k}{\sum_{k=1}^{S} \sigma^2_k} \tag{4.15}$$

where

$$\sigma^2_k = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left[\hat{\Delta}_k(\omega)\right]^2 - \left[\frac{1}{|\Omega|} \sum_{\omega \in \Omega} \hat{\Delta}_k(\omega)\right]^2 \tag{4.16}$$

with Ω containing the positions ω assumed by the parameter s_k,

$$\hat{\Delta}_k(\omega) = \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \bar{\Delta}(s), \qquad \mathcal{B} = \{s : s_k = \omega\} \tag{4.17}$$

and

$$\bar{\Delta}(s) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} [\Delta(s)]_p \tag{4.18}$$

Performance and data profiles

In order to evaluate the relative performance of DIRMIN and DIRMIN-2, the procedure proposed by [115] is also used. The following convergence condition is applied:

$$f(x^0) - f(x^n) \geq (1 - \tau)(f(x^0) - f_L) \tag{4.19}$$

where:

- $f(x^0)$ is the objective function value at the starting point, namely the function value at the center of the unit hyper-cube $\mathcal{U}$;
- $f(x^n)$ is the objective function value at the n-th evaluation;
- $0 \leq \tau \leq 1$ is a suitably chosen tolerance;
- f_L is the smallest function value obtained by any solver within the same maximum computational budget.

The main concepts needed to formally define data and performance profiles are recalled in the following. Let $\mathcal{A}$ be a set of $|\mathcal{A}|$ algorithms, and $\mathcal{P}$ a set of $|\mathcal{P}|$ problems and a performance measure $m_{p,a}$. In this work, $m_{p,a}$ is the number of function evaluations needed for algorithm a to satisfy (4.19) on problem p. The performance on problem p by algorithm a is compared with the best performance by any algorithm on this problem, using the following *performance ratio*:

$$r_{p,a} = \frac{m_{p,a}}{\min\{m_{p,a} \,:\, a \in \mathcal{A}\}} \tag{4.20}$$

Thus, a first measure of the performance of algorithm a is defined by the *performance profile*:

$$\rho_a(\alpha) = \frac{1}{|\mathcal{P}|} \text{size}\{p \in \mathcal{P} \,:\, r_{p,a} \leq \alpha\} \tag{4.21}$$

which approximates the probability for algorithm $a \in \mathcal{A}$ that the performance ratio $r_{p,a}$ is within a factor $\alpha \in \mathbb{R}$ of the best possible ratio. The convention $r_{p,a} = \infty$ is used when algorithm a fails to satisfy the convergence test (4.19) for problem $p \in \mathcal{P}$. We remark that performance profiles are plotted for values of the performance ration α such that

$$1 \leq \alpha \leq 1.1 \max_{p \in \mathcal{P}, a \in \mathcal{A}} r_{p,a}.$$

A further measure of the algorithms' performance is given by the percentage of problems that can be solved (for a given tolerance τ) within a certain number of ν simplex gradient evaluations. Namely, the so called *data profile* is defined as:

$$d_a(\nu) = \frac{1}{|\mathcal{P}|} \text{size}\{p \in \mathcal{P} \,:\, \frac{m_{p,a}}{n_p + 1} \leq \nu\} \tag{4.22}$$

where n_p is the number of variables in $p \in \mathcal{P}$. If the convergence test (4.19) cannot be satisfied within the assigned computational budget, $m_{p,a}$ is set equal to ∞. We again remark that data profiles are plotted for values of ν such that

$$0 \leq \nu \leq 1.1 \max_{p \in \mathcal{P}, a \in \mathcal{A}} \frac{m_{p,a}}{n_p + 1}.$$

4.2 Ship design problems

The following subsections present the ten SBD optimization applications pertaining the hull-form optimization of the high-speed Delft catamaran (DC) [15] and the USS Arleigh Burke-class destroyer (DTMB 5415) [116]. Table 4.7 summarized the algorithms applied on the problems described in the following subsections.

Table 4.7: SBDO problems

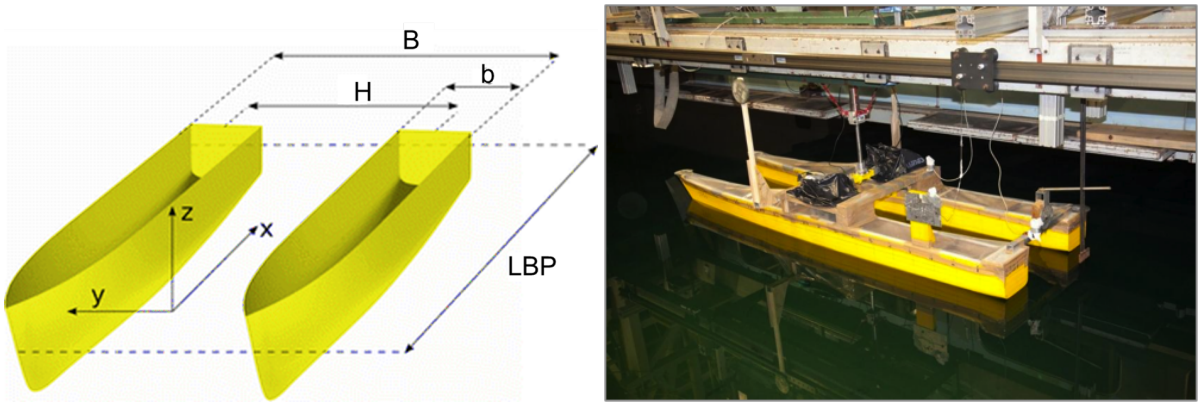
Problem	Hull	N	DPSO	LS-DF_PSO	DIRECT	DIRMIN	DIRMIN-2	FSA
I*	DC	4	✓	–	–	–	–	–
II*	DC	6	✓	–	–	–	–	–
III	DC	4	✓	✓	–	–	–	–
IV	DC	4	✓	✓	–	–	–	–
V	DTMB 5415	8	–	–	✓	✓	✓	–
VI	DTMB 5415	6	✓	✓	✓	–	✓	–
VII	DTMB 5415	11	✓	✓	✓	–	✓	–
VIII	DTMB 5415	1÷6	–	✓	–	–	–	–
IX	DTMB 5415	6	✓	–	✓	–	–	✓
X	DTMB 5415	6	✓	–	✓	–	–	✓

* In this case DPSO includes S-DPSO and A-DPSO, where is not specified DPSO stay for S-DPSO

4.2.1 Delft catamaran

The high-speed Delft catamaran (DC) [15] is an international benchmark geometry, which has been used for CFD and EFD studies. Earlier research on the Delft catamaran includes seakeeping CFD [117] and EFD [118], steady drift CFD [119] and EFD [120] for the onset and progression of vortices, CFD [121, 122] and EFD [123] interference factor studies, and CFD waterjet/maneuvering studies [124] and deterministic design optimization [16, 47, 125].

A sketch of the Delft catamaran model is shown in Fig. 4.5a. Figure 4.5b shows the DC model used for towing tank experiments at CNR-INSEAN. The hull main particulars and test conditions are summarized in Tab. 4.8.



(a) (b)

Figure 4.5: Delft catamaran (a) model sketch [26] and (b) model for the towing tank experiments at CNR-INSEAN

Four SBD problems (see Tab. 4.7) are used to study and verify the performance of the optimization algorithms. Specifically, problem I and II are used to assess S-DPSO and A-DPSO, whereas problem III and IV are used to compare DPSO with its global/local hybrid LS-DF_PSO. The problems details pertaining the hull-form optimization of the Delft catamaran are described in the following.

Table 4.8: Delft catamaran main particulars and simulation conditions

Main particular/condition	Symbol	Value, stochastic-speed	Value, stochastic-wave	Units
Length overall	LOA	3.822	105.4	m
Length between perpendiculars	LBP	3.627	100.0	m
Beam overall	B/LBP	0.313	0.313	–
Beam demi-hull	b/LBP	0.080	0.080	–
Draft at mid-ship	T/LBP	0.050	0.050	–
Distance between center of demi-hulls	H/LBP	0.234	0.234	–
Longitudinal center of gravity	LCG/LBP	0.527	0.527	–
Vertical center of gravity	VCG/LBP	0.074	0.113	–
Pitch radius of gyration	K_{yy}/LBP	not used	0.261	–
Froude number	Fr	[0.402;0.598]	0.5	–
Reynolds number	Re	$1.019e7$	$7.144e6$	–
Wave period	T_w	not used	[2.2;17.7]	s
Wave height	H_w	not used	[0.5;6.4]	m

Problem I

The single-objective optimization of DC [47] is formulated as:

$$\begin{array}{ll} \text{Minimize} & f = R_T(\mathbf{x})/W(\mathbf{x}) \\ \text{subject to} & LOA(\mathbf{x}) \leq LOA_0 \\ & B(\mathbf{x}) \leq B_0 \\ & T(\mathbf{x}) \leq T_0 \end{array} \tag{4.23}$$

where R_T is the total resistance at Fr=0.5 in calm water, and W is the weight force modulus. Geometry modifications have to fit in a box, defined by maximum overall length (LOA_0), beam (B_0) and draught (T_0), where subscript "0" is referred to the parent hull value. A four design variables space ($N = 4$) is used and includes overall dimension bounds. Modifications of the parent hull are performed using high-dimensional FFD (see Sec. 2.1) and 95%-confidence dimensionality reduction based on KLE (see Sec. 2.5) [16]. New designs $\mathbf{g}$ are produced as

$$\mathbf{g}(\mathbf{x}) = \left(1 - \sum_{i=1}^{N} x_i\right)\mathbf{g}_0 + \sum_{i=1}^{N} x_i\,\mathbf{g}_i \tag{4.24}$$

where $-1 \leq x_i \leq 1, \forall i \in [1,N]$ are the design variables; $\mathbf{g}_0$ is the original geometry and $\mathbf{g}_i$ are the geometries associated to the design space principal directions (or eigenmodes), as provided by KLE for dimensionality reduction.

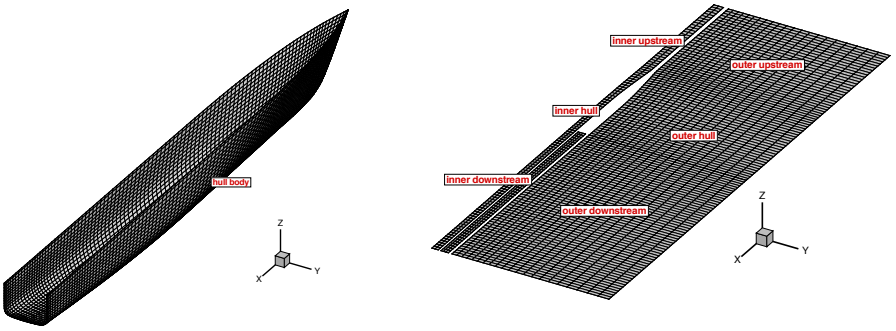


Figure 4.6: Panel-grid for the CNR-INSEAN WARP potential-flow code.

Simulations are conducted using the code WARP. Wave resistance computations are based on linear potential flow theory using NK linearization. The wave resistance is evaluated with a standard pressure integral, whereas the frictional resistance is estimated using a flat-plate approximation, based on the local Reynolds number [102]. The steady 2DOF (sinkage and trim) equilibrium is achieved by iteration of the flow solver and the body equation

of motion. Simulations are performed for the right demi-hull, since the problem is symmetrical with respect to the xz-plane. The free surface is discretized as follows: 20×1 panels on the inner-upstream sub-domain, 20×40 on the outer-upstream, 20×1 on the inner-hull, 20×40 on the outer-hull, 80×1 on the inner-dowstream, 80×2 on the transom-downstream, 80×40 on the outer-downstream; the body is discretized by 125×50 panels (see Fig. 4.6). Domain bounds are defined by $1LBP$ upstream, $4LBP$ downstream and $2LBP$ sideways.

Problem II

The single-objective optimization of DC is formulated as:

$$\begin{array}{rl} \text{Minimize} & f = R_T(\mathbf{x})/W(\mathbf{x}) \\ \text{subject to} & LBP(\mathbf{x}) = LBP_0 \\ \text{and to} & LOA(\mathbf{x}) \leq LOA_0 \\ & B(\mathbf{x}) \leq B_0 \\ & T(\mathbf{x}) \leq T_0 \end{array} \tag{4.25}$$

Differently from problem I, in this case a six design variables space ($N = 6$) is used, including constant length between perpendiculars as equality constraint. Subscript "0" is referred to the parent hull value. Geometry modification method, numerical solver and computational grid are the same of problem I.

Problem III

The design optimization of DC is formulated as:

$$\text{Minimize} \quad f = R_T(\mathbf{x})/\nabla(\mathbf{x}) \tag{4.26}$$

the minimization of the total resistance over displacement (∇) in calm water at Fr=0.5 [47]. A four design variables ($N = 4$) control global shape modifications, based on the KLE of the shape modification vector [16], is used. The problem is solved using a stochastic radial-basis functions metamodel (details may be found in Ref. [26]) trained by high-fidelity URANS simulations.

Problem IV

The design optimization of DC is formulated as:

$$\text{Minimize} \quad f = EV[\overline{R}_T(\mathbf{x})]_{SS,U} \tag{4.27}$$

the reduction of the expected value (EV) of the mean resistance ($\overline{R}_T$) in head waves, taking into account stochastic sea state (SS) in the North Pacific ocean and variable speed (U) [97]. A four design variables ($N = 4$) control global shape modifications, based on the KLE of the shape modification vector [16], is used. The problem is solved by means of stochastic radial-basis functions interpolation [26] of high-fidelity URANS simulations.

4.2.2 DTMB 5415

The SBD application presented is the hydrodynamic hull-form optimization of a USS Arleigh Burke-class destroyer, namely the DTMB 5415 model, an early and open to public version of the DDG-51. The DTMB 5415 model has been widely investigated through towing tank experiments [126, 127] and SBD studies, including hull-form optimization [128]. Recently, the DTMB 5415 model has been selected as the test case for the SBD activities within the NATO AVT-204, aimed at a multi-objective design optimization for multi-speed reduced resistance and improved seakeeping performance [116], and AVT-252 for stochastic design optimization.

Figure 4.7: A 5.720 m length model of the DTMB 5415 (CNR-INSEAN model 2340)

Figure 4.7 shows the geometry of the CNR-INSEAN 2340 model, a geosym replica of the DTMB 5415 model used for towing tank experiments, as seen in [126]. The main particulars of the full/model scale and tests conditions

are summarized in Tab. 4.9. Since no rudder is considered here, the length between perpendiculars (*LBP*) is calculated from the fore perpendicular to the transom bottom edge.

Table 4.9: DTMB 5415 main particulars and test conditions

Description	Symbol	Full scale	Model scale	Unit
Displacement	∇	8,636	0.549	tonnes
Length between perpendiculars	LBP	142.0	5.720	m
Beam	B	18.90	0.760	m
Draft	T	6.160	0.248	m
Longitudinal center of gravity	LCG	71.60	2.884	m
Vertical center of gravity	VCG	1.390	0.056	m
Roll radius of gyration	K_{xx}/B	0.400	0.400	–
Pitch radius of gyration	K_{yy}/LBP	0.250	0.250	–
Yaw radius of gyration	K_{zz}/LBP	0.250	0.250	–
Froude number	Fr	[0.250;0.410]*	0.250	–
Reynolds number	Re	1.215e9	9.824e6	–
Water density	ρ	998.5	998.5	kg/m^3
Kinematic viscosity	ν	1.09e-6	1.09e-6	m^2/s
Gravity acceleration	g	9.803	9.803	m/s^2
Wave period	T_w	9.700	not used	s
Wave height	H_w	3.250	not used	m

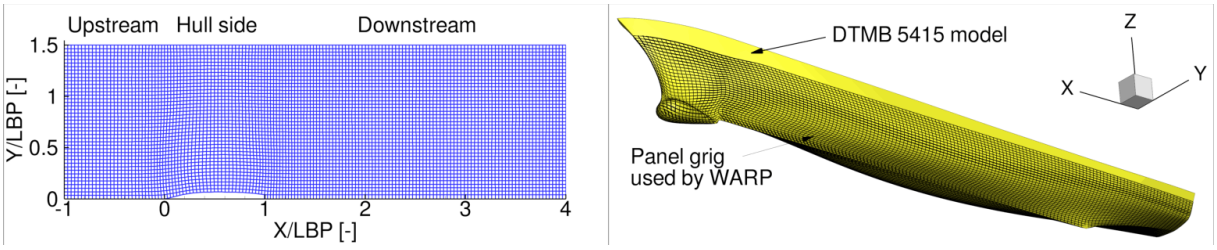
* Fr=0.41 is used only for seakeeping optimization

Six SBD problems (see Tab. 4.7) are used to study and verify the performance of the optimization algorithms. Specifically, problem V is used to compare the performance of DIRECT and its global/local hybridization, problem VI and VII compare DPSO and DIRECT to LS-DF_PSO and DIRMIN-2, problem VIII is used to show the benefits of dimensionality reduction applying LS-DF_PSO as optimization algorithm, and finally problems IX and X compare the performance of DPSO, DIRECT and FSA. The problems details pertaining the hull-form optimization of the DTMB 5415 are described in the following subsections.

Low- and high-fidelity simulations are performed for the right demi-hull, taking advantage of symmetry about the xz-plane. The potential flow (PF) computational domain for the free surface is defined within $1LBP$ upstream, $3LBP$ downstream and $1.5LBP$ sideways, as shown in Fig. 4.8a. The associated panel grid used (Fig. 4.8b) is summarized in Tab. 4.10 and guarantee solution convergence. Table 4.11 summarizes the associated background and boundary layer volume grids used for the RANS solver, designed to have $y^+ = 0.3$ at Fr=0.25. The computational domain and grids are shown in Figs. 4.9.

Table 4.10: Computationel panel grid for PF solver (WARP)

Hull	Free surface			Total
	Upstream	Hull side	Downstream	
150 × 30	30 × 44	30 × 44	90 × 44	11k

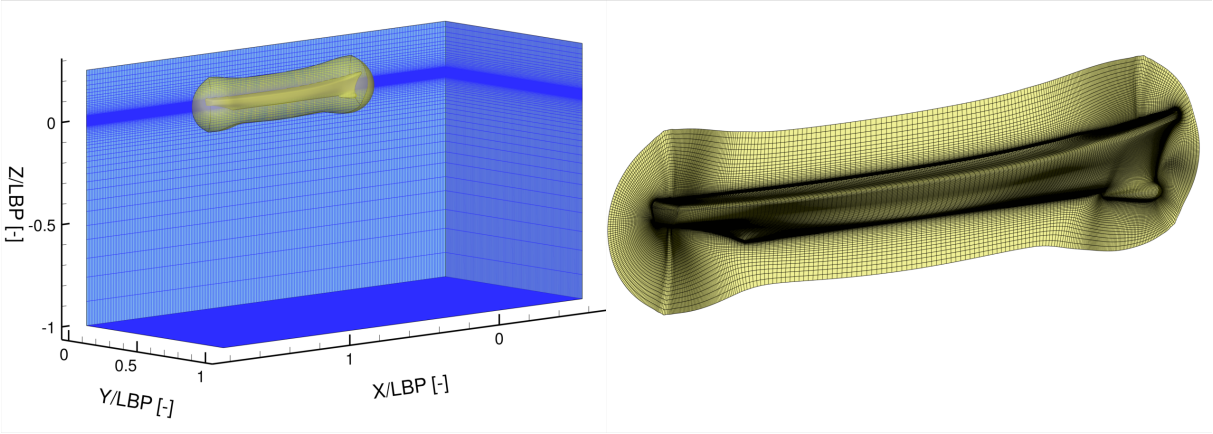


(a) Free-surface (b) Hull

Figure 4.8: Computational panel grid for WARP

Table 4.11: Computational volume grid for RANS solver (CFDShip-Iowa)

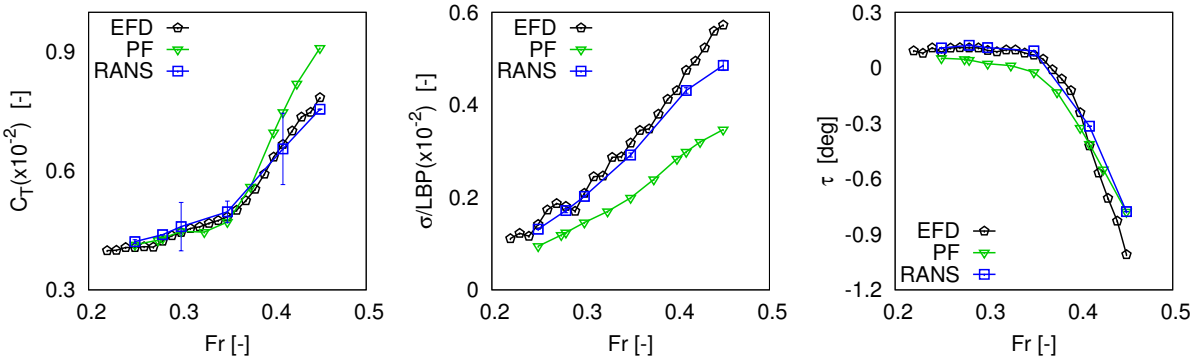
Background	Boundary layer	Total
$227 \times 155 \times 115$	$243 \times 71 \times 115$	6M



(a) Background (b) Boundary layer

Figure 4.9: Computational volume grid for CFDShip-Iowa

The validation of the PF and RANS analyses performed by WARP and CFDShip-Iowa, respectively, for the original hull is shown in Fig. 4.10 versus experimental (EFD) data collected at CNR-INSEAN [129], showing a reasonable agreement especially for low speeds; $C_T = R_T/0.5\rho U^2 S_{w,stat}$, σ/LBP, and τ are shown, where U is the undisturbed flow speed, $S_{w,stat}$ is the static wetted surface area, σ is the sinkage (positive if the center of gravity sinks), and τ is the trim angle (positive if the bow sinks). Error bars indicate the grid uncertainty, evaluated using the factor of safety method [130].



(a) (b) (c)

Figure 4.10: Total resistance coefficient $C_T = R_T/0.5\rho U^2 S_{w,stat}$ (a), non-dimensional sinkage (b), and trim (c) in calm water versus Fr, for the model scale DTMB 5415 (LBP=5.72 m)

Problem V

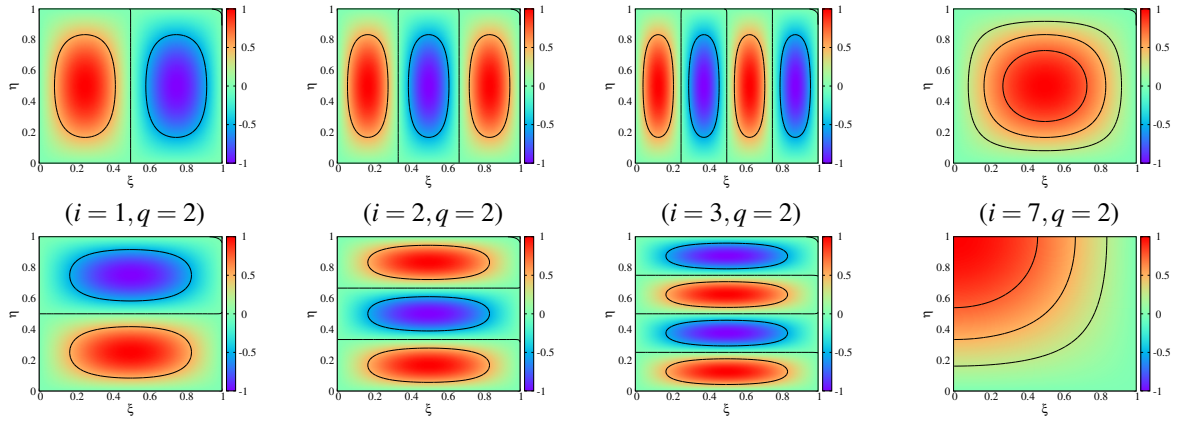
The single-objective optimization of DTMB 5415 is formulated as:

$$\begin{aligned} \text{Minimize} \quad & f = R_T(\mathbf{x}) \\ \text{subject to} \quad & LBP(\mathbf{x}) = LBP_0 \\ & \nabla(\mathbf{x}) = \nabla_0 \\ \text{and to} \quad & |\Delta B(\mathbf{x})| \leq 0.05 B_0 \\ & |\Delta T(\mathbf{x})| \leq 0.05 T_0 \end{aligned} \tag{4.28}$$

where R_T is the total resistance in calm water at Fr equal to 0.25 in full scale. Geometrical constraints include fixed length between perpendiculars (LBP) and fixed displacement (∇), with beam (B) and draft (T) varying between $\pm 5\%$ of the original value. The OBFE_2D $\boldsymbol{\psi}_i \in \mathbb{R}^3$ $(j = 1, ..., 8)$ (see Sec. 2.2) are applied for the modification of the hull shape, controlled by $N = 8$ design variables $\alpha_i \in \mathbb{R}$ $(i = 1, \ldots, 8)$, as

$$\boldsymbol{\delta}_s(\xi, \eta) = \sum_{i=1}^{8} \alpha_i \, \boldsymbol{\psi}_i(\xi, \eta) \tag{4.29}$$

Specifically, six orthogonal functions and design variables are used for the hull, whereas two functions/variables are used for the sonar dome. The corresponding functions are shown in Fig. 4.11. Table 4.12 summarized the parameters used herein, including upper and lower bounds used for the dimensional design variables α_i. The results will be presented in the following in terms of non-dimensional design variables $x_i \in [-1, 1]$.



$(i = 1, q = 2)$ $(i = 2, q = 2)$ $(i = 3, q = 2)$ $(i = 7, q = 2)$

$(i = 4, q = 2)$ $(i = 5, q = 2)$ $(i = 6, q = 2)$ $(i = 8, q = 3)$

(a) Hull modification (b) Sonar dome modification

Figure 4.11: Problem V, 2D orthogonal functions $\boldsymbol{\psi}_i(\xi, \eta)$

Table 4.12: Problem V, 2D orthogonal functions parameters, for shape modification

Description	i	r_i	ϕ_i	t_i	χ_i	$q(i)$	$\alpha_{i,\min}$ [m]	$\alpha_{i,\max}$ [m]
Hull modification	1	2.0	0	1.0	0	2	-1.0	1.0
	2	3.0	0	1.0	0	2	-1.0	1.0
	3	4.0	0	1.0	0	2	-1.0	1.0
	4	1.0	0	2.0	0	2	-0.5	0.5
	5	1.0	0	3.0	0	2	-0.5	0.5
	6	1.0	0	4.0	0	2	-0.5	0.5
Sonar dome modification	7	1.0	0	1.0	0	2	-0.3	0.3
	8	0.5	$\pi/2$	0.5	0	3	-0.5	0.5

The solver used is the potential flow code WARP, based on the NK linearization [99]. The wave resistance is evaluated with the transverse wave cut method [101], whereas the friction resistance is estimated by a local approximation based on flat-plate theory [102]. The steady 2DOF (sinkage and trim) equilibrium is achieved by iteration of the flow solver and the body equation of motion.

Problem VI

The single-objective optimization of DTMB 5415 is formulated as:

$$\begin{aligned} \text{Minimize} \quad & f = R_T(\mathbf{x}) \\ \text{subject to} \quad & LBP(\mathbf{x}) = LBP_0 \\ & \nabla(\mathbf{x}) = \nabla_0 \\ \text{and to} \quad & |\Delta B(\mathbf{x})| \leq 0.05 B_0 \\ & |\Delta T(\mathbf{x})| \leq 0.05 T_0 \end{aligned} \tag{4.30}$$

where R_T is the total resistance in calm water at Fr equal to 0.25 in full scale. Geometrical constraints include fixed length between perpendiculars (LBP) and fixed displacement (∇), with beam (B) and draft (T) varying between $\pm 5\%$ of the original value.

Shape modifications $\boldsymbol{\delta}_s$ are produced directly on the Cartesian coordinates $\mathbf{x}_s$ of the computational body surface grid, as per

$$\mathbf{x}_s(\boldsymbol{\alpha}) = \mathbf{x}_{s,0} + \boldsymbol{\delta}_s(\boldsymbol{\alpha}) \tag{4.31}$$

where $\boldsymbol{\alpha}$ is the design variable vector and $\mathbf{x}_{s,0}$ represents the original body surface grid. The shape modification $\boldsymbol{\delta}_s$ is defined using $N = 6$ OBFE_2D (see Sec. 2.2) of the curvilinear coordinates ξ and η over the (demi) hull

$$\boldsymbol{\psi}_i(\xi,\eta) \; : \; \mathcal{G} = [0, L_\xi] \times [0, L_\eta] \in \mathbb{R}^2 \longrightarrow \mathbb{R}^3, \qquad i = 1, \ldots, 6 \tag{4.32}$$

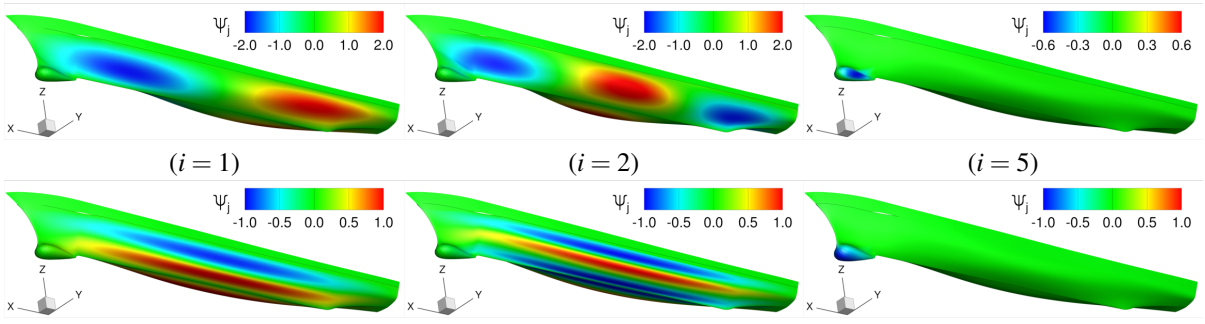
as

$$\boldsymbol{\delta}_s(\xi,\eta) = \sum_{i=1}^{6} \alpha_i \, \boldsymbol{\psi}_i(\xi,\eta) \tag{4.33}$$

Four functions and design variables are used for the hull, whereas two functions/variables are used for the sonar dome, as summarized in Tab. 4.13. The corresponding basis functions are shown in Fig. 4.12. Upper and lower bounds used for the design variables α_i are included in Tab. 4.13.

Table 4.13: Problem VI, OBFE_2D parameters for shape modification

Description	i	r_i	ϕ_i	t_i	χ_i	$q(i)$	$\alpha_{i,\min}$ [m]	$\alpha_{i,\max}$ [m]
Hull modification	1	2.0	0	1.0	0	2	-2.0	2.0
	2	3.0	0	1.0	0	2	-2.0	2.0
	3	1.0	0	2.0	0	2	-1.0	1.0
	4	1.0	0	3.0	0	2	-1.0	1.0
Sonar dome modification	5	1.0	0	1.0	0	2	-0.6	0.6
	6	0.5	$\pi/2$	0.5	0	3	-1.0	1.0



($i = 1$) ($i = 2$) ($i = 5$)

($i = 3$) ($i = 4$) ($i = 6$)

(a) Hull modification (b) Sonar dome modification

Figure 4.12: Problem VI, 2D orthogonal functions $\boldsymbol{\psi}_i(\xi,\eta)$

Fixed LBP and ∇ are satisfied by automatic geometric scaling, while constraints for B and T are handled using a penalty function method. This is used here, since the relationship between beam/draft variations and design variables is not explicitly provided by the orthogonal expansion and geometric scaling.

Simulations are conducted using the code WARP. Wave resistance computations are based on linear potential flow theory using NK linearization. The wave resistance is evaluated with the transverse wave cut method [101, 131], whereas the frictional resistance is estimated using a flat-plate approximation, based on the local Reynolds number [102]. The steady 2DOF (sinkage and trim) equilibrium is achieved by iteration of the flow solver and the body equation of motion.

For the hull form optimization process, a limit to the number of function evaluations is set equal to 1536, i.e. $256N$.

Problem VII

The single-objective optimization of DTMB 5415 is formulated as:

$$\begin{array}{rl} \text{Minimize} & f = C_T(\mathbf{x}) \\ \text{subject to} & LBP(\mathbf{x}) = LBP_0 \\ \text{and to} & |\Delta B(\mathbf{x})| \leq 0.05 B_0 \\ & |\Delta T(\mathbf{x})| \leq 0.05 T_0 \end{array} \tag{4.34}$$

where C_T is the total resistance coefficient in calm water at Fr equal to 0.25 in model scale. Geometrical constraints include fixed length between perpendiculars (LBP), with beam (B) and draft (T) varying between $\pm 5\%$ of the original value.

The shape modification $\boldsymbol{\delta}_s$ is defined using $M = 27$ OBFE_3D of the Cartesian coordinates x, y, z over a hyper-rectangle

$$\boldsymbol{\varphi}_i(x,y,z) \; : \; \mathcal{G} = [0,L_x] \times [0,L_y] \times [0,L_z] \in \mathbb{R}^3 \longrightarrow \mathbb{R}^3, \qquad j = i, ..., M \tag{4.35}$$

as

$$\boldsymbol{\delta}_s(x,y,z) = \sum_{i=1}^{M} \beta_i \, \boldsymbol{\varphi}_i(x,y,z) \tag{4.36}$$

where the coefficients $\beta_i \in \mathbb{R}$ $(i = 1, \ldots, M)$ are the design variables, and $\boldsymbol{\varphi}_i$ are the OBFE_3D (see Sec. 2.3). Table 4.14 summarizes the parameters used herein.

Table 4.14: Problem VII, OBFE_3D parameters for shape modification

i	n_i	ϕ_i	m_i	χ_i	l_i	θ_i	$q(i)$	$\beta_{i,\min}$ [m]	$\beta_{i,\max}$ [m]
1	1.0	0	1.0	0	1.0	0	2	-1.0	1.0
2	1.0	0	1.0	0	2.0	0	2	-1.0	1.0
3	1.0	0	2.0	0	1.0	0	2	-1.0	1.0
4	2.0	0	1.0	0	1.0	0	2	-1.0	1.0
5	1.0	0	2.0	0	2.0	0	2	-1.0	1.0
6	2.0	0	1.0	0	2.0	0	2	-1.0	1.0
7	2.0	0	2.0	0	1.0	0	2	-1.0	1.0
8	2.0	0	2.0	0	2.0	0	2	-1.0	1.0
9	1.0	0	1.0	0	3.0	0	2	-1.0	1.0
10	1.0	0	3.0	0	1.0	0	2	-1.0	1.0
11	3.0	0	1.0	0	1.0	0	2	-1.0	1.0
12	1.0	0	2.0	0	3.0	0	2	-1.0	1.0
13	2.0	0	1.0	0	3.0	0	2	-1.0	1.0
14	1.0	0	3.0	0	2.0	0	2	-1.0	1.0
15	2.0	0	3.0	0	1.0	0	2	-1.0	1.0
16	3.0	0	1.0	0	2.0	0	2	-1.0	1.0
17	3.0	0	2.0	0	1.0	0	2	-1.0	1.0
18	2.0	0	2.0	0	3.0	0	2	-1.0	1.0
19	2.0	0	3.0	0	2.0	0	2	-1.0	1.0
20	3.0	0	2.0	0	2.0	0	2	-1.0	1.0
21	1.0	0	3.0	0	3.0	0	2	-1.0	1.0
22	3.0	0	1.0	0	3.0	0	2	-1.0	1.0
23	3.0	0	3.0	0	1.0	0	2	-1.0	1.0
24	2.0	0	3.0	0	3.0	0	2	-1.0	1.0
25	3.0	0	2.0	0	3.0	0	2	-1.0	1.0
26	3.0	0	3.0	0	2.0	0	2	-1.0	1.0
27	3.0	0	3.0	0	3.0	0	2	-1.0	1.0

The design space defined by Eq. 4.36 is reduced in dimensionality using the generalized KLE method presented Sec. 2.5 and in Ref. [16]:

$$\boldsymbol{\delta}_s(\xi,\eta) \approx \sum_{i=1}^{N} \alpha_i \boldsymbol{\Phi}_i(\xi,\eta) \tag{4.37}$$

where ξ and η are curvilinear coordinates over the (demi) hull, $\alpha_i \in \mathbb{R}$ $(i = 1,\dots,N)$ are new design variables and $\boldsymbol{\Phi}_i$ are the solutions of the eigenproblem

$$\iint_{\mathcal{G}} \rho(\xi\prime,\eta\prime)\,\langle \boldsymbol{\delta}_s(\xi,\eta)\otimes\boldsymbol{\delta}_s(\xi\prime,\eta\prime)\rangle \\ \boldsymbol{\Phi}_i(\xi\prime,\eta\prime)d\xi\prime d\eta\prime = \lambda_i\boldsymbol{\Phi}_i(\xi,\eta) \tag{4.38}$$

provided that $\langle \boldsymbol{\delta}_s(\xi,\eta,\boldsymbol{\alpha})\rangle = \mathbf{0}$, $\forall (x,y,z) \in \mathcal{G}$. The brackets $\langle\cdot\rangle$ indicate ensemble average over the realizations of the design variable vector $\boldsymbol{\beta} = \{\beta_i\} \in \mathcal{D}$, which is assessed here using 10,000 uniform random samples (each of them resulting in a modified hull form). The weight $\rho \in \mathbb{R}^+$ defines a generalized inner product and is used to give more emphasis to submerged grid nodes. The following orthogonality property holds [132, 133]

$$\iint_{\mathcal{G}} \rho(\xi,\eta)\boldsymbol{\Phi}_i(\xi,\eta)\cdot\boldsymbol{\Phi}_j(\xi,\eta)dxdydz = \delta_{ij} \tag{4.39}$$

Finally, the reduced dimension N is selected in order to retain the 90% of the original geometric variance, as

$$\sum_{i=1}^{N}\lambda_i \geq 0.90\sum_{k=1}^{\infty}\lambda_k \tag{4.40}$$

provided that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_i \geq \lambda_{i+1}$. For the current problem, $N = 11$ and the corresponding eigenfunctions (represented on the hull) are shown in Fig. 4.13. Details of the formulation and numerical implementation of the design space dimensionality reduction technique may be found in [16]. Details of the application to the hull form optimization of the DTMB 5415 may be found in [86].

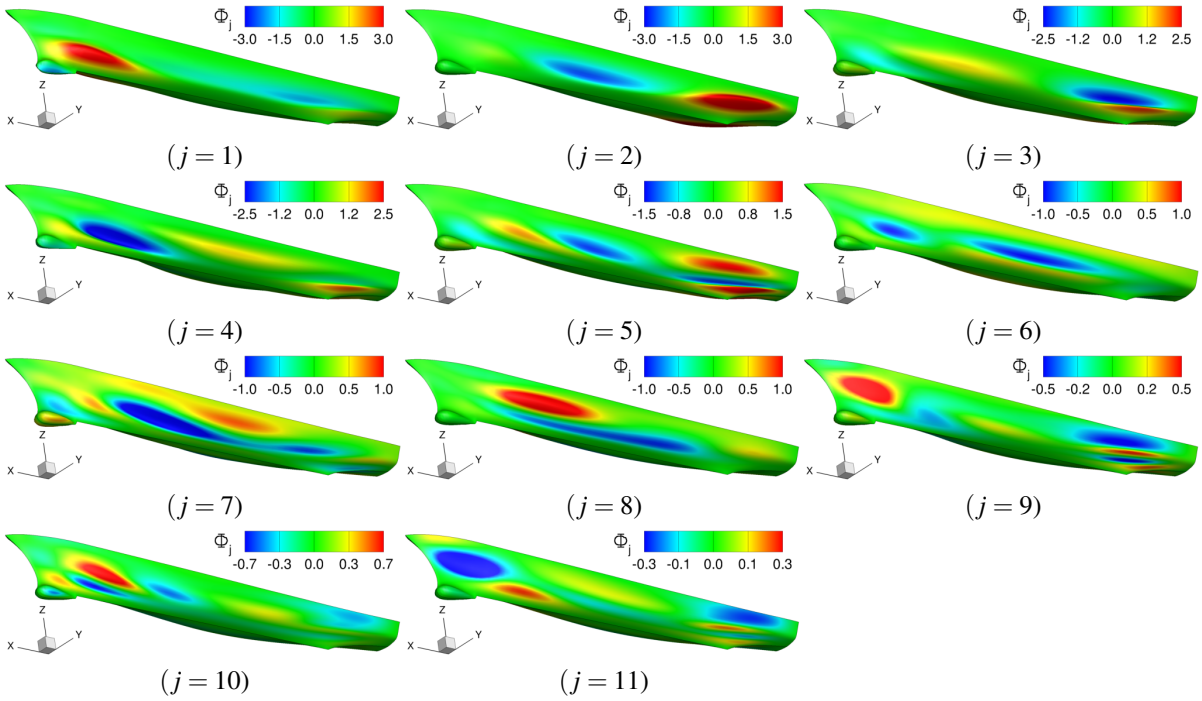


Figure 4.13: Problem VII, KLE solutions $\boldsymbol{\Phi}_j(x,y,z)$

The boundary layer grid is modified as presented in Sec. 2.4.1, setting $c = 0$ in Eq. 2.17 and $l = \bar{l}$ in Eq. 2.16. During the metamodel training, each modified grid is assessed for quality by means of y^+, determinant of the Jacobian matrix and skewness.

Simulations are performed with the RANS code CFDShip-Iowa V4.5 [104]. The SST blended $k-\varepsilon/k-\omega$ turbulent model is used. The free-surface location is predicted by a single phase level set method. A second order upwind scheme is used to discretize the convective terms of momentum equations. For a high performance parallel computing, an MPI-based domain decomposition approach is used, where each decomposed block is mapped to one processor. The code SUGGAR runs as a separate process from the flow solver to compute interpolation coefficients for the overset grid, which enables CFDShip-Iowa to take care of 6DOF with a motion controller at every time step. Only 2DOFs are considered in the current study.

The optimization problem is solved using a metamodel (see Appendix B), trained by 71 RANS simulations defined using a sequential sampling procedure [27]. The metamodel used is a first order polyharmonic spline, which is a special case of radial basis function (RBF) interpolation [26].

For the hull form optimization process, a limit to the number of metamodel-based function evaluations is set equal to 8800, i.e. $800N$.

Problem VIII

A single-objective shape optimization is formulated as:

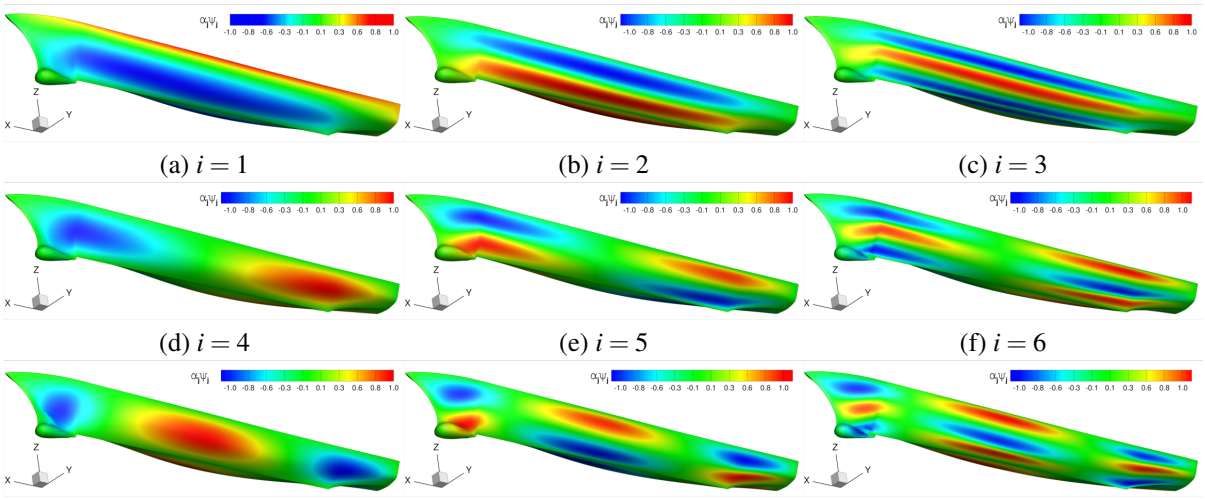
$$\begin{array}{rl} \text{Minimize} & f = R_T(\mathbf{x}) \\ \text{subject to} & LBP(\mathbf{x}) = LBP_0 \\ & \nabla(\mathbf{x}) = \nabla_0 \\ \text{and to} & |\Delta B(\mathbf{x})| \leq 0.05 B_0 \\ & |\Delta T(\mathbf{x})| \leq 0.05 T_0 \\ & V(\mathbf{x}) \geq V_0 \end{array} \tag{4.41}$$

where $\mathbf{x}$ is the design variable vector, R_T is the resistance in calm water at 18 kn (corresponding to Fr = 0.25), LBP is the length between perpendiculars, ∇ is the displacement, B is the beam, T is the draft, and V is the volume reserved for the sonar in the dome. Subscript '0' indicates parent hull values.

A linear expansion of the shape modification vector $\boldsymbol{\delta}_s \in \mathbb{R}^3$ is used [90]. Specifically, $M = 9$ OBFE_2D $\boldsymbol{\psi}_i \in \mathbb{R}^3$ $(i = 1,...,M)$ are applied for the modification of the hull shape (see Sec. 2.2), controlled by nine design variables $\alpha_i \in \mathbb{R}$ $(i = 1,\ldots,M)$, as

$$\boldsymbol{\delta}_s(\xi,\eta) = \sum_{i=1}^{M} \alpha_j \, \boldsymbol{\psi}_i(\xi,\eta) \tag{4.42}$$

The corresponding functions are shown in Fig. 4.14. Table 4.15 summarized the parameters used for the current study.



(a) $i = 1$ (b) $i = 2$ (c) $i = 3$

(d) $i = 4$ (e) $i = 5$ (f) $i = 6$

(g) $i = 7$ (h) $i = 8$ (i) $i = 9$

Figure 4.14: Problem VIII, 2D orthogonal functions $\boldsymbol{\psi}_i(\xi,\eta)$

The equality constraints for LBP and ∇ are automatically satisfied by the geometry modification tool, through scaling along x (for LBP), y and z (for ∇). This introduces nonlinearities in the representation of Eq. 4.42. Inequality (geometric) constraints are treated by a penalty function method.

After design-space dimensionality reduction is performed, shape modification are produced directly on the computational grid $\mathbf{g}$, so that

$$\mathbf{g}(\alpha_1,\ldots,\alpha_N) = \mathbf{g}_0 + \sum_{i=1}^{N} \sup\{\alpha_i\} \boldsymbol{\varphi}_i \tag{4.43}$$

where g is the modified-shape grid, $\mathbf{g}_0$ is the grid associated to the mean shape (corresponding to the original hull) and α_i are the new design variable. The term $\sup\{\alpha_i\}$ is used only for normalization.

Table 4.15: Problem VIII, OBFE_2D parameters for shape modification

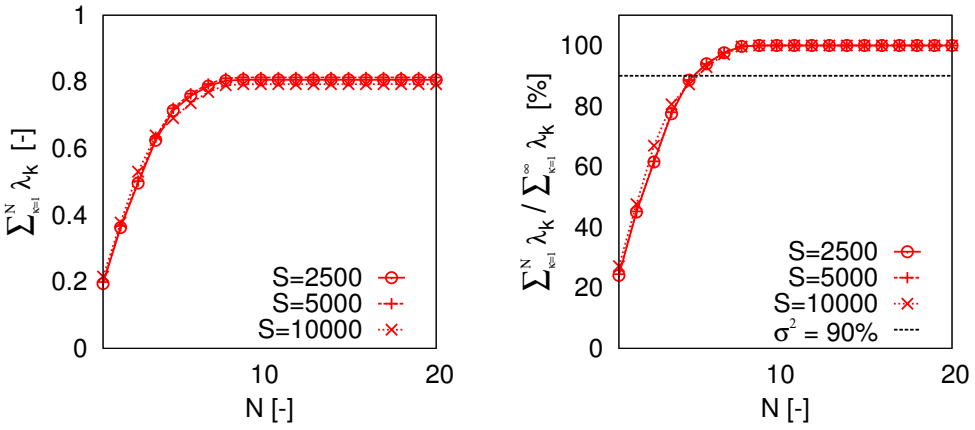
i	r_i	ϕ_i	t_i	χ_i	$q(i)$
1	1.0	0	1.0	0	2
2	1.0	0	2.0	0	2
3	1.0	0	3.0	0	2
4	2.0	0	1.0	0	2
5	2.0	0	2.0	0	2
6	2.0	0	3.0	0	2
7	3.0	0	1.0	0	2
8	3.0	0	2.0	0	2
9	3.0	0	3.0	0	2

The weight function $\rho(\mathbf{x})$, for the dimensionality reduction procedure, is defined so as to give more emphasis to submerged nodes (in water), so that

$$\begin{cases} \rho(\mathbf{x}) = 1, & \text{if } z \leq z_{WL} \\ \rho(\mathbf{x}) = 0, & \text{if } z > z_{WL} \end{cases} \tag{4.44}$$

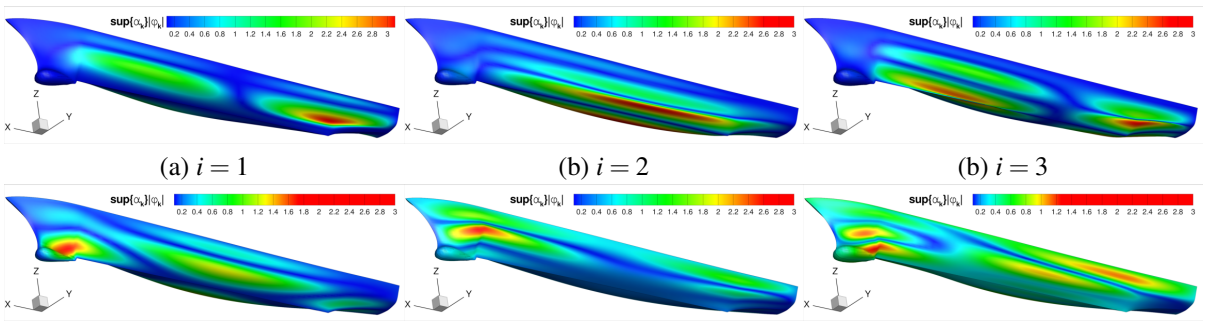
where z_{WL} represent the location of the water line. The design space define by the $M = 9$ orthogonal function is sampled using a uniform random distribution of $S = 2500$, 5000, and 10000 items. The KL values cumulative sum, (corresponding to the geometric variance retained by a N-dimensional design space), is shown in Fig. 4.15a. Results are found convergent versus S. The reduced-dimensionality spaces, are formed by choosing $N = 1, \ldots, 6$, retaining approximately the 27%, 48%, 67%, 81%, 87% and 92% of the original geometric variance, respectively. The KL modes and the corresponding hull shapes are shown in Fig. 4.16.

Simulations are conducted using the code WARP (Wave Resistance Program). Wave resistance computations are based on the DM linear PF theory [100]. The wave resistance is evaluated with the standard pressure integral, whereas the frictional resistance is estimated using a flat-plate approximation, based on the local Reynolds number [102].



(a) (b)

Figure 4.15: Problem VIII, geometric variance retained by a reduced-dimensionality space of dimension N: (a) absolute and (b) percent values



(a) $i = 1$ (b) $i = 2$ (b) $i = 3$

(d) $k = i$ (e) $i = 5$ (f) $i = 6$

Figure 4.16: Problem VIII, KL modes ($i = 1, \ldots, 6$)

Problem IX

A single-objective shape optimization is formulated as:

$$\begin{array}{rl} \text{Minimize} & f = \dfrac{R_T}{\nabla}\dfrac{\nabla_0}{R_{T_0}} \\ \text{subject to} & LBP(\mathbf{x}) = LBP_0 \\ \text{and to} & |\Delta B(\mathbf{x})| \leq 0.05 B_0 \\ & |\Delta T(\mathbf{x})| \leq 0.05 T_0 \\ & V(\mathbf{x}) \geq V_0 \end{array} \tag{4.45}$$

where, the objective function is the normalized total resistance R_T over displacement ∇ ratio in calm water at 18 kn (Fr=0.25). Geometrical equality constraints include fixed length between perpendicular (LBP), whereas geometrical inequality constraints include limited variation of beam and draught ($\pm 5\%$) and reserved volume (V) for the sonar in the dome, corresponding to 4.9 m diameter and 1.7 m length (cylinder). A $N = 6$ design space is used, corresponding to that used and described for Problem VIII.

The SBDO problem is solved using the code WARP. Wave resistance computations are based on the DM linear PF theory [100]. The wave resistance is evaluated with the standard pressure integral, whereas the frictional resistance is estimated using a flat-plate approximation, based on the local Reynolds number [102].

Problem X

A single-objective shape optimization is formulated as:

$$\begin{array}{rl} \text{Minimize} & f = 0.5\,\dfrac{RMS(a_z)}{RMS(a_{z_0})}\Big|_0^{30kn} + 0.5\,\dfrac{RMS(\varphi)}{RMS(\varphi_0)}\Big|_{150}^{18kn} \\ \text{subject to} & LBP(\mathbf{x}) = LBP_0 \\ \text{and to} & |\Delta B(\mathbf{x})| \leq 0.05 B_0 \\ & |\Delta T(\mathbf{x})| \leq 0.05 T_0 \\ & V(\mathbf{x}) \geq V_0 \end{array} \tag{4.46}$$

where RMS represents the root mean square, a_z is the vertical acceleration of the bridge (located 27 m forward amidships and 24.75 m above keel) at 30 kn (Fr=0.41) in head wave (0 deg), and φ is the roll angle at 18 kn (Fr=0.25) in stern long-crested wave (150 deg). The wave conditions corresponds to sea state 5, using the Bretschneider spectrum with a significant wave height of 3.25 m and modal period of 9.7 s. Subscript '0' is referred to the parent hull value.

Geometrical equality constraints include fixed length between perpendicular, whereas geometrical inequality constraints include limited variation of beam and draught ($\pm 5\%$) and reserved volume for the sonar in the dome, corresponding to 4.9 m diameter and 1.7 m length (cylinder). A $N = 6$ design space is used, corresponding to that used and described for Problem VIII and IX.

Simulations are performed using the code SMP, that provides a potential flow solution based on linearized strip theory. The 6DOF response is given, advancing at constant forward speed with arbitrary heading in both regular waves and irregular seas, as well as the longitudinal, lateral, and vertical responses at specified location of the ship [103]. The associated hull grid is the same used for the other DTMB 5415 problems described before, and the validation versus the experimental data collected at the IIHR [134] is shown in Fig. 4.17, showing a reasonable agreement.

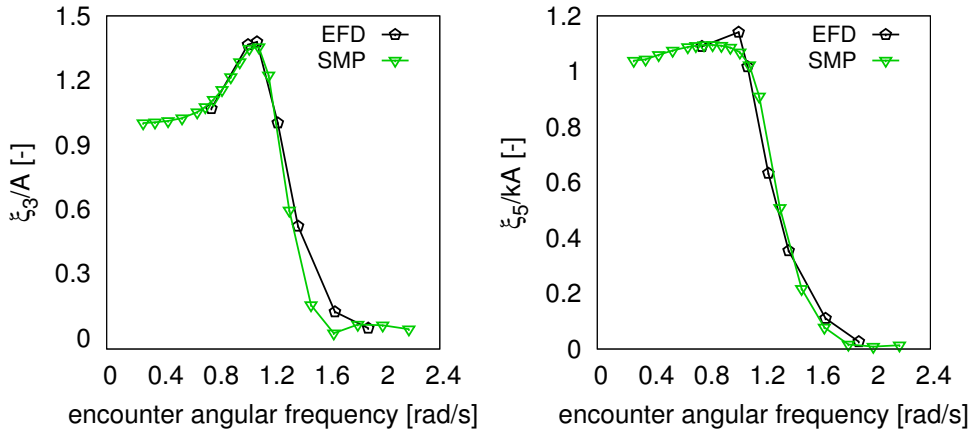


Figure 4.17: Problem X, seakeeping solver validation in head waves at 30 kn

Chapter 5

Numerical results

"Where should I go" - Alice."That depends on where you want to end up" - The Cheshire Cat.

— Lewis Carroll, *Alice in Wonderland*

5.1 Analytical test functions

Test-function results are presented in the following and used to define the guidelines for DPSO, DIRMIN, DIRMIN-2, and FSA adopted later for the SBD problem.

5.1.1 DPSO

Figures 5.1 and 5.3, show the performances of SD-PSO versus the budget of function evaluations, in terms of Δ_x, Δ_f, Δ_t, for $N < 10$ and ≥ 10 respectively. Average values are presented, conditional to number of particles, particles initialization, coefficient set and wall-type approach, respectively.

Figures 5.2 and 5.4 show the relative variance σ_r^2 of Δ_x, Δ_f, Δ_t for $N < 10$ and ≥ 10 respectively, retained by each of the aforementioned parameter. The particles initialization is found the most significant parameter to affect S-DPSO performance, especially for $N \geq 10$, whereas the coefficient set (selected herein) and the wall-type approach are shown to be the least important.

Tables 5.1 and 5.2 show the five best performing setups for each Δ_x, Δ_f, and Δ_t, for $N < 10$ and ≥ 10 respectively, varying the budget of function evaluations $N_{\max}$. Average values and standard deviations (STD) for all S-DPSO setups are also provided.

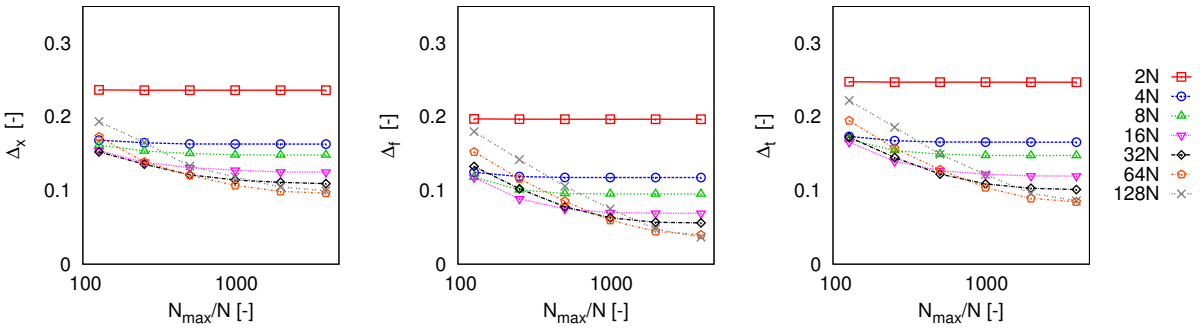
Generally, A-DPSO results are found similar to S-DPSO. Specifically, Figs. 5.5 and 5.7, show the performances of A-DPSO versus the budget of function evaluations, in terms of Δ_x, Δ_f, Δ_t, for $N < 10$ and ≥ 10 respectively. Average values are presented, conditional to number of particles, particles initialization, coefficient set and wall-type approach respectively.

Figures 5.6 and 5.8 show the relative variance σ_r^2 of Δ_x, Δ_f, Δ_t for $N < 10$ and ≥ 10 respectively, retained by each of the aforementioned parameters. The particles initialization is again the most significant parameter to affect A-DPSO performance, especially for low budgets and $N \geq 10$. The coefficient set (selected herein) and the wall-type approach are shown to have a limited effect on the performance, compared to other parameters.

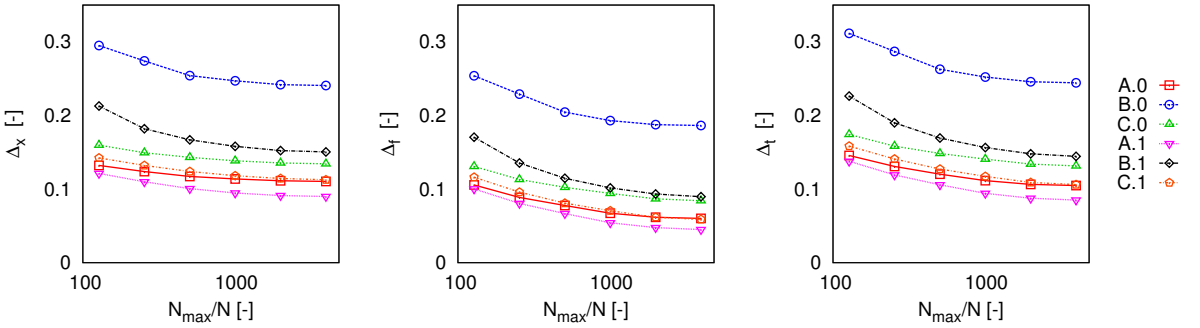
Tables 5.3 and 5.4 summarize the five best performing setups for each Δ_x, Δ_f, and Δ_t, for $N < 10$ and ≥ 10, respectively, varying the budget of function evaluations. Overall averages and standard deviations for all A-DPSO setups are also included

The most frequent setups are selected from Tabs. 5.1, 5.2, 5.3 and 5.4, in order to define a reasonable and robust guideline for the use of S-DPSO and A-DPSO. These are summarized in Tab. 5.5.

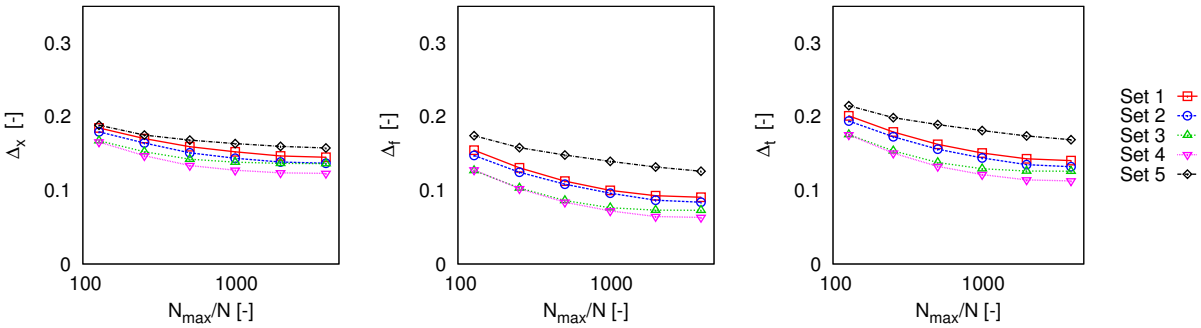
Figures 5.9 and 5.10 show the performance of the suggested setups, for S-DPSO and A-DPSO and $N < 10$ and ≥ 10, respectively. Average performance, standard deviation and best performing setup among all combinations is also shown for each budget. The guideline setups ("Guide") are found always very close or coincident to the "Best". In addition, it may be noted how A-DPSO is always equivalent or slightly better than S-DPSO.



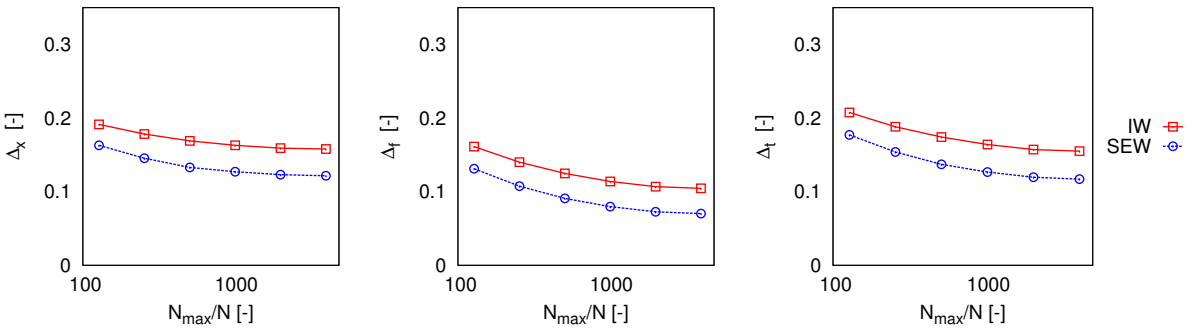
(a) Average performance, conditional to swarm size N_p



(b) Average performance, conditional to swarm initialization

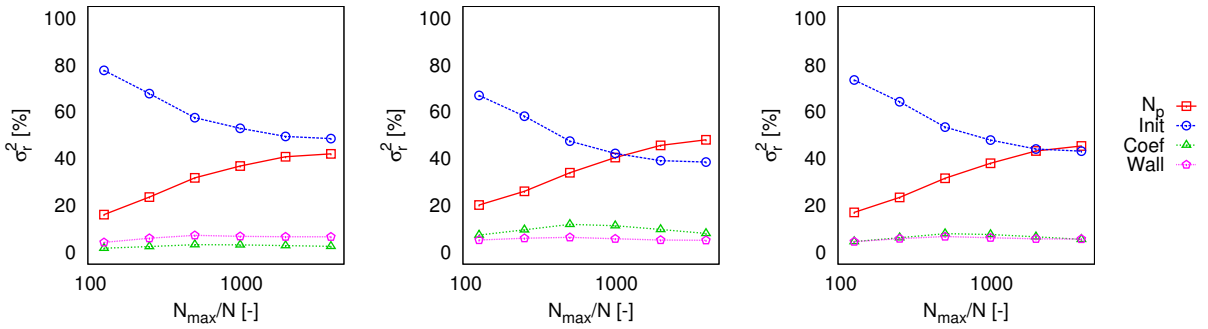


(c) Average performance, conditional to coefficient set



(d) Average performance, conditional to wall-type approach

Figure 5.1: S-DPSO average performance for $N < 10$

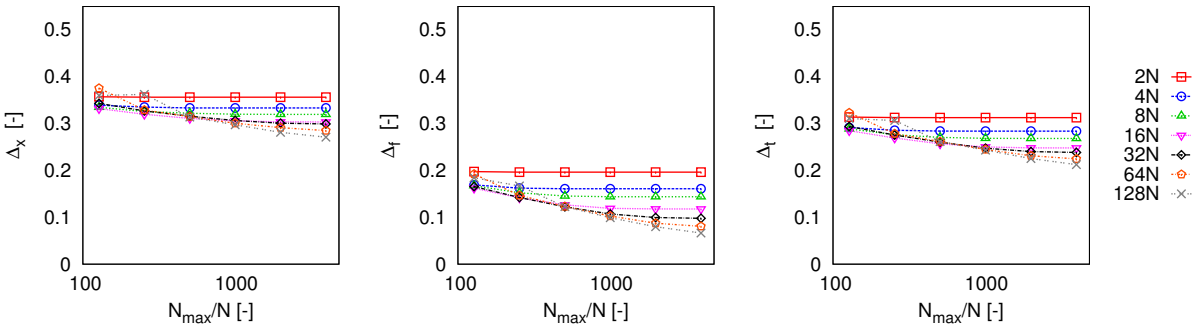


(a) Δ_x (b) Δ_f (c) Δ_t

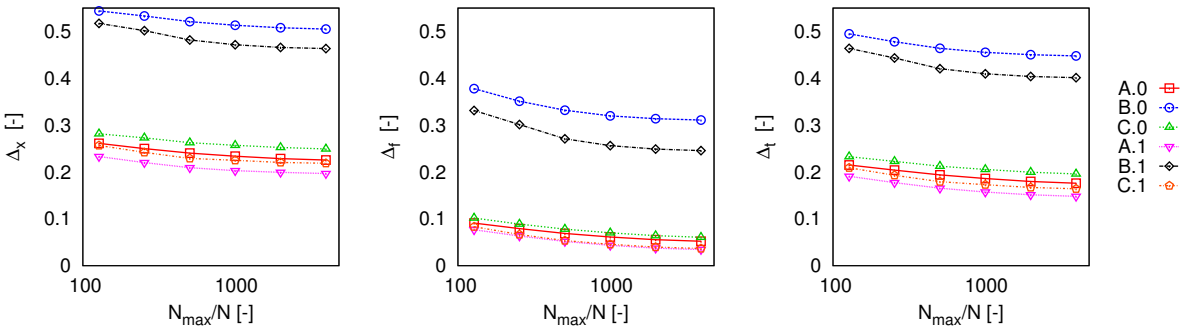
Figure 5.2: Relative variance $\sigma_r^2(\%)$ of S-DPSO setting parameters for $N < 10$

Table 5.1: Best performing setups for S-DPSO, $N < 10$

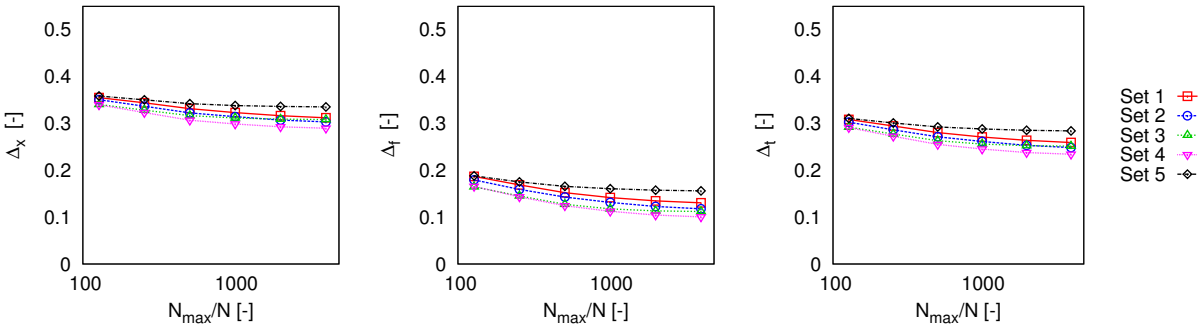
N_{max}/N	Average (STD)			Best S-DPSO													
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_t
				4	C.1	4	SEW	0.081	4	C.1	4	IW	0.015	4	C.1	4	SEW	0.064
	0.177	0.146	0.192	4	C.1	3	IW	0.084	2	A.1	4	SEW	0.017	4	C.1	4	IW	0.066
128				4	C.1	4	IW	0.086	4	C.0	4	SEW	0.017	2	A.1	4	SEW	0.069
	(0.080)	(0.084)	(0.085)	32	A.1	3	SEW	0.088	4	C.1	4	SEW	0.018	4	C.0	4	SEW	0.076
				16	A.1	3	SEW	0.089	2	A.1	4	IW	0.030	2	A.1	4	IW	0.078
				32	A.1	3	SEW	0.071	4	C.1	4	SEW	0.013	4	C.1	4	SEW	0.060
	0.162	0.124	0.171	64	A.1	3	SEW	0.071	4	C.1	4	IW	0.014	4	C.1	4	IW	0.062
256				4	C.1	4	SEW	0.078	4	C.0	4	SEW	0.015	16	A.1	3	SEW	0.066
	(0.077)	(0.084)	(0.084)	16	A.1	3	SEW	0.079	2	A.1	4	SEW	0.017	2	A.1	4	SEW	0.069
				4	C.1	4	IW	0.081	16	A.1	3	SEW	0.024	4	C.0	4	SEW	0.072
				64	A.1	4	IW	0.053	4	C.1	4	SEW	0.013	32	A.1	3	SEW	0.055
	0.151	0.108	0.156	64	A.1	2	IW	0.060	4	C.1	4	IW	0.014	64	A.1	3	IW	0.057
512				32	A.1	3	SEW	0.062	4	C.0	4	SEW	0.015	64	A.1	3	SEW	0.058
	(0.077)	(0.085)	(0.085)	64	A.1	3	IW	0.063	2	A.1	4	SEW	0.017	4	C.1	4	SEW	0.059
				64	A.1	3	SEW	0.064	32	A.1	3	SEW	0.018	4	C.1	4	IW	0.061
				64	A.1	4	IW	0.047	64	A.1	3	SEW	0.002	64	A.1	3	SEW	0.035
	0.145	0.097	0.145	64	A.1	3	SEW	0.049	32	A.1	3	SEW	0.002	64	A.1	4	IW	0.036
1024				64	A.1	2	IW	0.053	64	A.0	4	IW	0.003	64	A.1	3	IW	0.041
	(0.079)	(0.088)	(0.089)	64	A.1	2	SEW	0.055	64	A.1	3	IW	0.004	32	A.1	3	SEW	0.042
				64	A.1	3	IW	0.057	32	A.0	4	SEW	0.004	64	A.1	4	SEW	0.046
				64	A.1	2	IW	0.042	64	A.1	3	SEW	0.001	64	A.1	2	IW	0.030
	0.141	0.090	0.138	64	A.1	4	IW	0.045	64	A.1	4	SEW	0.001	64	A.1	4	IW	0.032
2048				64	A.1	3	SEW	0.047	64	A.1	4	IW	0.001	64	A.1	3	SEW	0.033
	(0.080)	(0.091)	(0.091)	64	A.1	2	SEW	0.053	64	A.0	4	SEW	0.001	64	A.1	2	SEW	0.038
				64	A.1	3	IW	0.055	64	A.0	4	IW	0.002	64	A.1	3	IW	0.039
				64	A.1	2	IW	0.039	128	A.1	3	IW	0.000	64	A.1	2	IW	0.027
	0.140	0.087	0.136	64	A.1	4	IW	0.044	128	C.1	3	SEW	0.001	64	A.1	4	IW	0.031
4096				64	A.1	3	SEW	0.047	64	A.1	2	IW	0.001	64	A.1	3	SEW	0.033
	(0.081)	(0.092)	(0.092)	64	A.1	2	SEW	0.050	128	C.1	4	SEW	0.001	64	A.1	2	SEW	0.035
				128	C.1	4	IW	0.051	64	C.0	4	SEW	0.001	128	C.1	4	IW	0.036
				64	A.1	4	IW	0.064	4	C.1	4	SEW	0.013	4	C.1	4	SEW	0.060
	0.153	0.109	0.156	64	A.1	3	SEW	0.065	4	C.1	4	IW	0.014	32	A.1	3	SEW	0.060
Av.				64	A.1	2	IW	0.065	4	C.0	4	SEW	0.015	4	C.1	4	IW	0.062
	(0.077)	(0.084)	(0.085)	32	A.1	3	SEW	0.066	2	A.1	4	SEW	0.017	64	A.1	3	SEW	0.066
				64	A.1	3	IW	0.070	32	A.1	3	SEW	0.023	64	A.1	4	IW	0.069



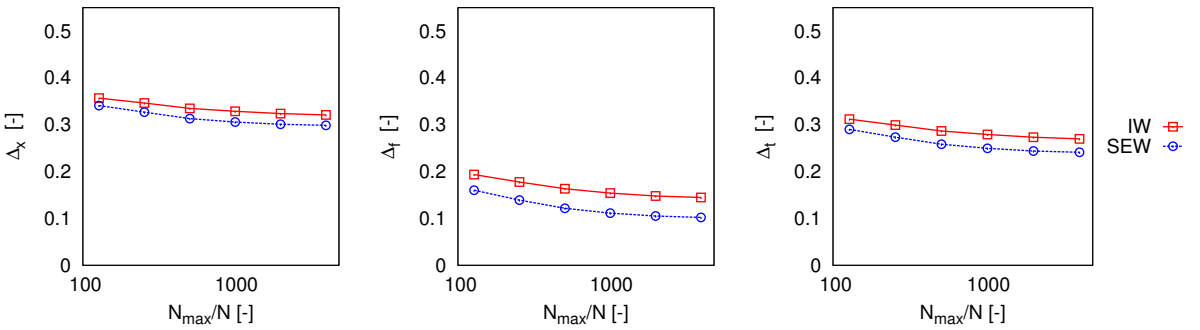
(a) Average performance, conditional to swarm size N_p



(b) Average performance, conditional to swarm initialization

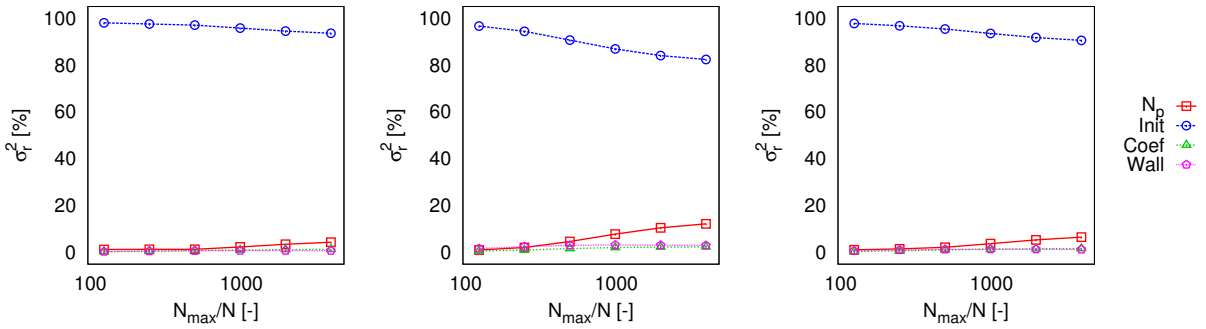


(c) Average performance, conditional to coefficient set



(d) Average performance, conditional to wall-type approach

Figure 5.3: S-DPSO average performance for $N \geq 10$

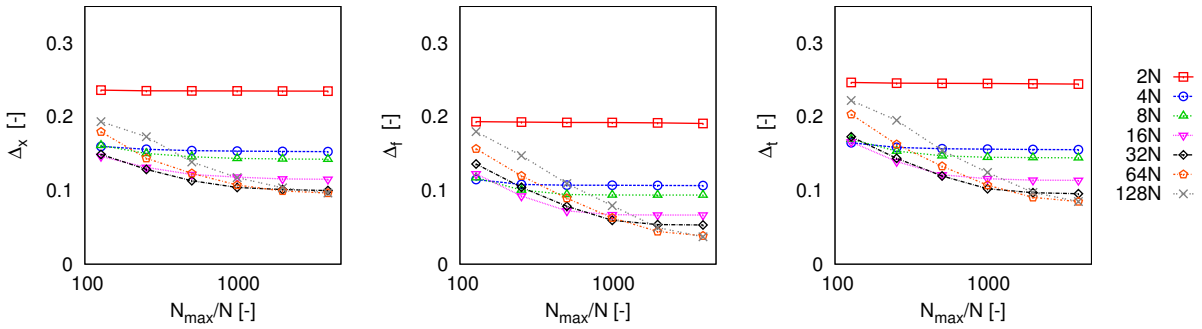


(a) Δ_x (b) Δ_f (c) Δ_t

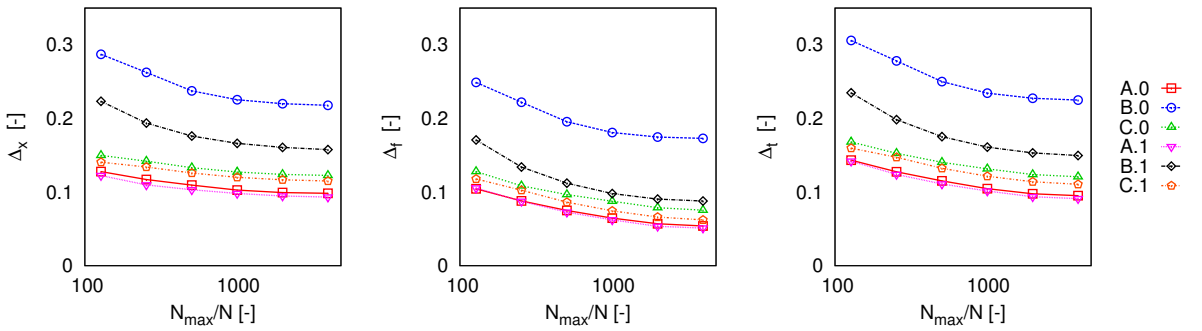
Figure 5.4: Relative variance $\sigma_r^2(\%)$ of S-DPSO setting parameters for $N \geq 10$

Table 5.2: Best performing setups for S-DPSO, $N \geq 10$

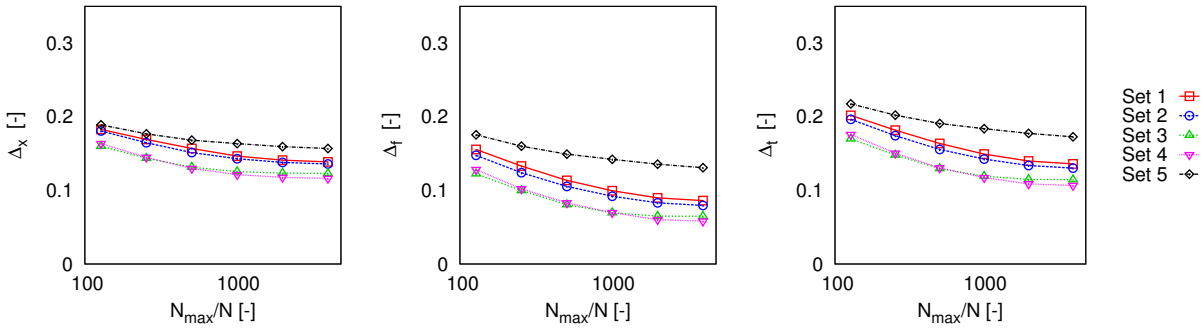
N_{max}/N	Average (STD)			Best S-DPSO													
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_t
				16	A.1	3	SEW	0.181	4	A.1	4	SEW	0.024	4	A.1	4	SEW	0.143
	0.349	0.177	0.301	2	A.1	4	SEW	0.192	4	A.1	4	IW	0.028	2	A.1	4	SEW	0.147
128				16	A.1	2	IW	0.193	8	A.1	3	SEW	0.034	16	A.1	3	SEW	0.150
	(0.134)	(0.137)	(0.133)	4	A.1	4	SEW	0.197	8	A.1	3	IW	0.035	4	A.1	4	IW	0.151
				4	C.1	4	SEW	0.198	4	A.0	3	SEW	0.036	2	A.1	2	IW	0.151
				16	A.1	3	SEW	0.174	4	A.1	4	SEW	0.016	16	A.1	3	SEW	0.136
	0.337	0.158	0.286	16	A.1	4	SEW	0.181	8	A.1	3	SEW	0.017	8	C.1	4	SEW	0.138
256				16	A.1	4	IW	0.182	4	A.1	4	IW	0.020	4	A.1	4	SEW	0.139
	(0.134)	(0.134)	(0.132)	32	A.1	3	SEW	0.187	8	A.1	3	IW	0.021	8	A.1	2	IW	0.139
				8	A.1	2	IW	0.190	8	C.1	4	SEW	0.022	4	C.1	4	SEW	0.143
				16	A.1	3	SEW	0.170	16	A.1	3	SEW	0.008	16	A.1	3	SEW	0.122
	0.324	0.143	0.273	16	A.1	4	SEW	0.173	8	C.1	4	SEW	0.015	16	A.1	4	IW	0.130
512				16	A.1	4	IW	0.173	8	A.1	3	SEW	0.015	16	A.1	4	SEW	0.131
	(0.133)	(0.133)	(0.131)	32	A.1	3	SEW	0.174	4	A.1	4	SEW	0.016	32	A.1	3	SEW	0.131
				32	A.1	4	SEW	0.176	16	A.1	3	IW	0.016	8	C.1	4	SEW	0.132
				32	A.1	4	SEW	0.164	32	A.1	3	SEW	0.004	16	A.1	3	SEW	0.120
	0.317	0.133	0.264	16	A.1	3	SEW	0.168	16	A.1	3	SEW	0.005	32	A.1	3	SEW	0.120
1024				16	A.1	4	SEW	0.168	32	C.1	3	SEW	0.007	16	A.1	2	IW	0.123
	(0.133)	(0.134)	(0.132)	32	A.1	3	SEW	0.168	64	A.1	3	SEW	0.011	32	A.1	4	SEW	0.123
				16	A.1	4	IW	0.170	32	A.1	3	IW	0.013	16	A.1	4	IW	0.123
				32	A.1	4	SEW	0.160	32	A.1	3	SEW	0.003	32	A.1	4	SEW	0.116
	0.312	0.127	0.259	16	A.1	2	IW	0.164	64	A.1	3	SEW	0.004	16	A.1	2	IW	0.117
2048				64	A.1	4	IW	0.165	32	C.1	3	SEW	0.005	32	A.1	3	SEW	0.118
	(0.134)	(0.136)	(0.133)	32	A.1	3	SEW	0.166	16	A.1	3	SEW	0.005	16	A.1	3	SEW	0.120
				16	A.1	3	SEW	0.168	64	C.1	3	IW	0.010	16	C.1	2	IW	0.120
				32	A.1	4	SEW	0.159	64	A.1	3	SEW	0.002	32	A.1	4	SEW	0.115
	0.310	0.123	0.256	64	A.1	4	IW	0.160	32	A.1	3	SEW	0.003	64	A.1	4	IW	0.115
4096				128	A.1	4	SEW	0.163	32	C.1	3	SEW	0.005	16	A.1	2	IW	0.117
	(0.134)	(0.137)	(0.135)	16	A.1	2	IW	0.163	16	A.1	3	SEW	0.005	32	A.1	3	SEW	0.118
				32	A.1	3	SEW	0.166	64	C.1	3	IW	0.008	128	A.1	4	SEW	0.118
				16	A.1	3	SEW	0.171	4	A.1	4	SEW	0.017	16	A.1	3	SEW	0.128
	0.325	0.143	0.273	32	A.1	4	SEW	0.177	16	A.1	3	SEW	0.017	32	A.1	3	SEW	0.136
Av.				16	A.1	2	IW	0.177	8	A.1	3	SEW	0.019	16	A.1	2	IW	0.136
	(0.133)	(0.133)	(0.132)	16	A.1	4	IW	0.178	8	C.1	4	SEW	0.020	8	C.1	4	SEW	0.137
				32	A.1	3	SEW	0.178	4	A.1	4	IW	0.021	16	A.1	4	IW	0.137



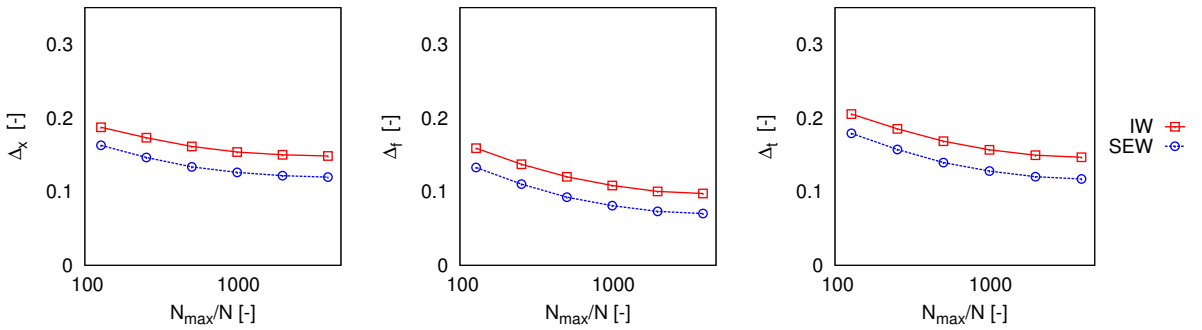
(a) Average performance, conditional to swarm size N_p



(b) Average performance, conditional to swarm initialization

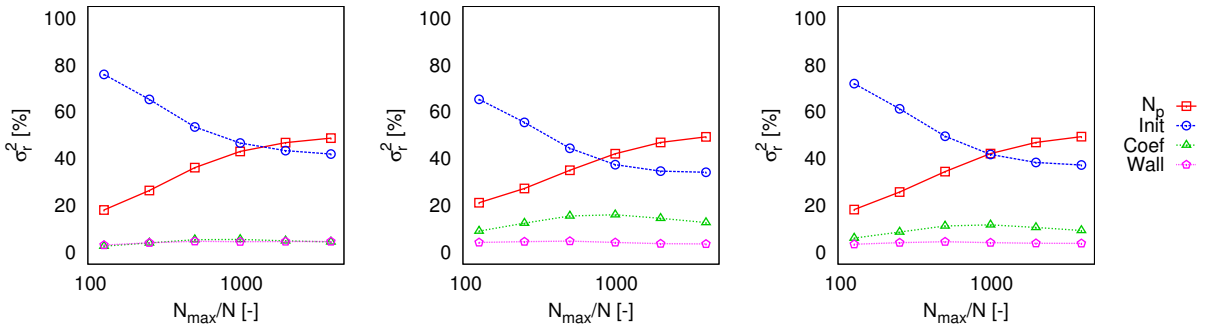


(c) Average performance, conditional to coefficient set



(d) Average performance, conditional to wall-type approach

Figure 5.5: A-DPSO average performance for $N < 10$

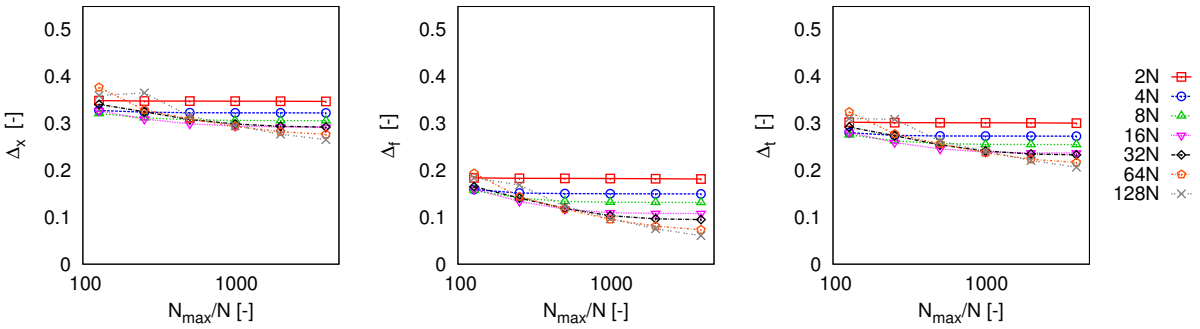


(a) Δ_x (b) Δ_f (c) Δ_t

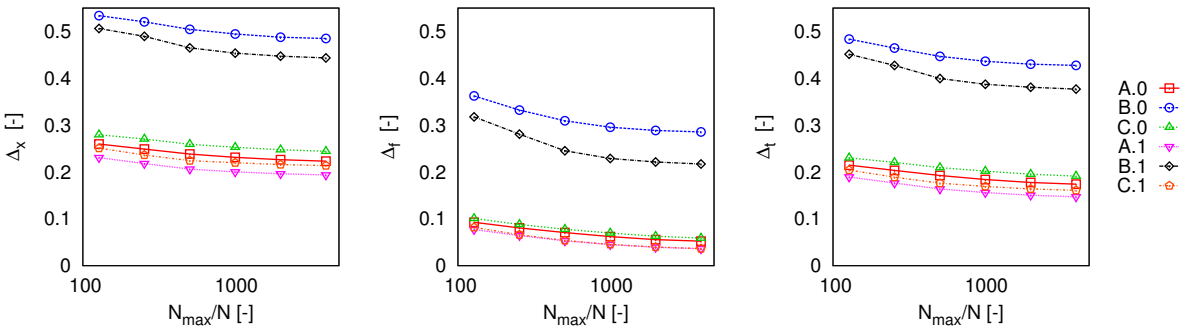
Figure 5.6: Relative variance $\sigma_r^2(\%)$ of A-DPSO setting parameters for $N < 10$

Table 5.3: Best performing setups for A-DPSO, $N < 10$

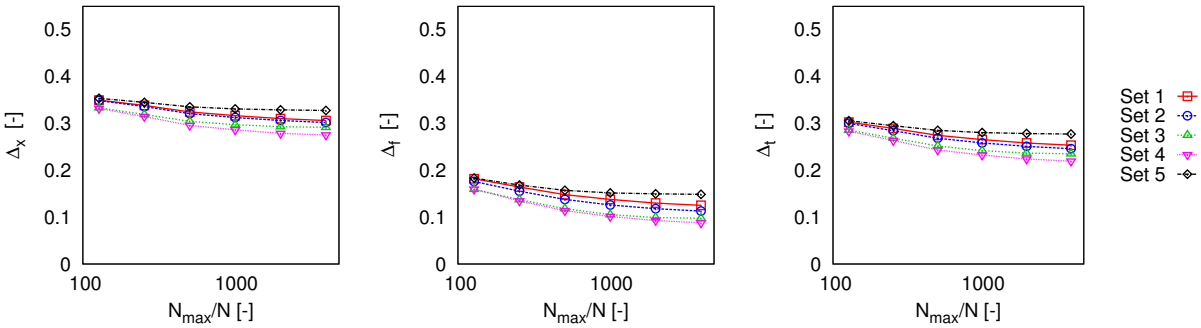
N_{max}/N	Average (STD)			Best A-DPSO													
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_t
				4	C.1	4	SEW	0.076	4	C.1	3	IW	0.004	4	C.1	4	IW	0.058
	0.175	0.146	0.192	4	C.1	4	IW	0.081	4	C.1	4	IW	0.005	4	C.1	4	SEW	0.062
128				8	A.1	3	IW	0.085	4	C.0	3	IW	0.015	4	C.1	3	IW	0.069
	(0.080)	(0.082)	(0.084)	2	A.0	4	SEW	0.092	4	C.0	4	SEW	0.021	4	C.0	3	IW	0.074
				16	A.1	3	SEW	0.092	4	C.0	4	IW	0.022	4	C.0	4	SEW	0.075
				4	C.1	4	IW	0.072	4	C.1	4	IW	0.002	4	C.1	4	IW	0.051
	0.160	0.124	0.171	4	C.1	4	SEW	0.074	4	C.1	3	IW	0.004	4	C.1	4	SEW	0.060
256				16	A.1	2	SEW	0.076	4	C.0	3	IW	0.014	4	C.0	4	SEW	0.066
	(0.076)	(0.081)	(0.082)	16	A.0	4	IW	0.077	4	C.0	4	IW	0.014	4	C.1	3	IW	0.067
				32	A.1	3	SEW	0.078	4	C.0	4	SEW	0.017	4	C.0	3	IW	0.071
				32	A.1	4	IW	0.064	4	C.1	4	IW	0.001	4	C.1	4	IW	0.046
	0.148	0.106	0.154	4	C.1	4	IW	0.065	4	C.1	3	IW	0.004	16	A.0	4	IW	0.051
512				32	C.1	4	IW	0.066	16	A.0	4	IW	0.005	4	C.1	4	SEW	0.059
	(0.074)	(0.082)	(0.081)	32	A.1	2	IW	0.069	4	C.0	4	IW	0.014	32	A.1	3	SEW	0.063
				32	A.0	4	SEW	0.070	4	C.0	3	IW	0.014	16	A.1	3	SEW	0.065
				32	A.1	4	IW	0.055	4	C.1	4	IW	0.001	4	C.1	4	IW	0.044
	0.140	0.095	0.142	64	A.1	4	SEW	0.059	16	A.0	4	IW	0.002	32	A.1	4	IW	0.044
1024				32	A.1	2	IW	0.061	4	C.1	3	IW	0.004	64	A.1	3	SEW	0.045
	(0.075)	(0.084)	(0.084)	32	A.0	4	SEW	0.061	64	A.1	3	SEW	0.005	16	A.0	4	IW	0.049
				32	C.1	4	IW	0.061	64	A.0	3	SEW	0.005	64	A.0	4	SEW	0.051
				64	A.1	4	SEW	0.051	4	C.1	4	IW	0.001	64	A.1	4	SEW	0.037
	0.136	0.087	0.135	32	A.1	4	IW	0.055	32	C.1	4	SEW	0.002	32	A.1	2	IW	0.040
2048				32	A.1	2	IW	0.056	16	A.0	4	IW	0.002	64	A.1	3	SEW	0.042
	(0.076)	(0.087)	(0.087)	64	A.1	3	SEW	0.059	64	A.0	4	IW	0.003	32	A.1	4	IW	0.043
				64	A.1	1	IW	0.059	32	A.1	2	IW	0.003	4	C.1	4	IW	0.044
				64	A.1	4	SEW	0.051	128	A.1	4	SEW	0.001	64	A.1	4	SEW	0.036
	0.134	0.084	0.132	32	A.1	2	IW	0.055	4	C.1	4	IW	0.001	32	A.1	2	IW	0.039
4096				32	A.1	4	IW	0.055	32	A.1	2	IW	0.002	64	A.1	1	IW	0.040
	(0.077)	(0.088)	(0.088)	64	A.1	1	IW	0.057	32	C.1	4	SEW	0.002	64	A.1	3	SEW	0.041
				64	A.1	3	SEW	0.057	16	A.0	4	IW	0.002	32	A.1	4	IW	0.043
				4	C.1	4	IW	0.067	4	C.1	4	IW	0.002	4	C.1	4	IW	0.048
	0.149	0.107	0.155	32	A.1	4	IW	0.070	4	C.1	3	IW	0.004	4	C.1	4	SEW	0.060
Av.				4	C.1	4	SEW	0.073	4	C.0	3	IW	0.014	4	C.0	4	SEW	0.067
	(0.074)	(0.081)	(0.081)	32	C.1	4	IW	0.074	4	C.0	4	IW	0.015	4	C.1	3	IW	0.067
				32	A.1	2	IW	0.074	4	C.0	4	SEW	0.018	16	A.0	4	IW	0.069



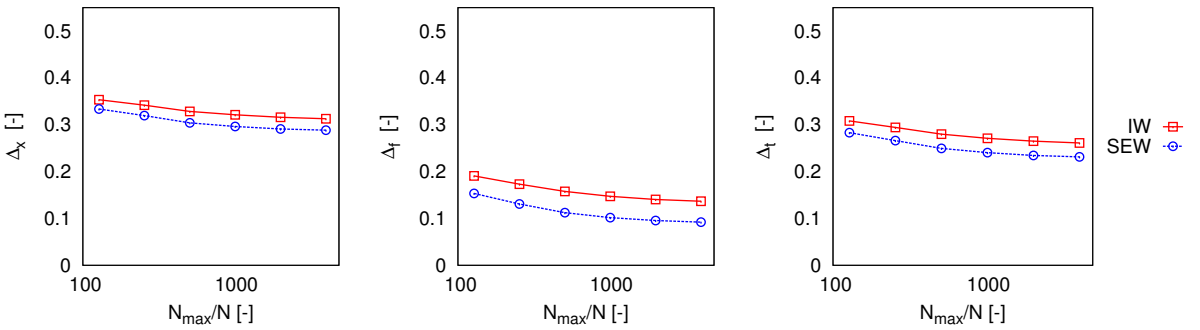
(a) Average performance, conditional to swarm size N_p



(b) Average performance, conditional to swarm initialization

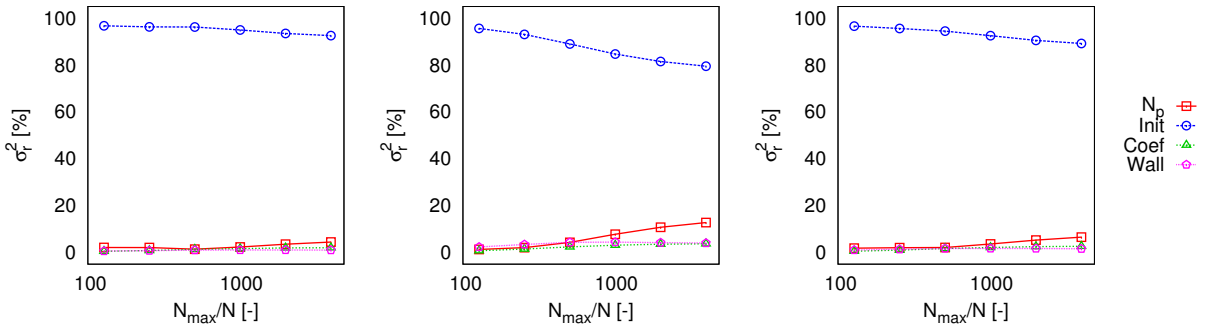


(c) Average performance, conditional to coefficient set



(d) Average performance, conditional to wall-type approach

Figure 5.7: A-DPSO average performance for $N \geq 10$

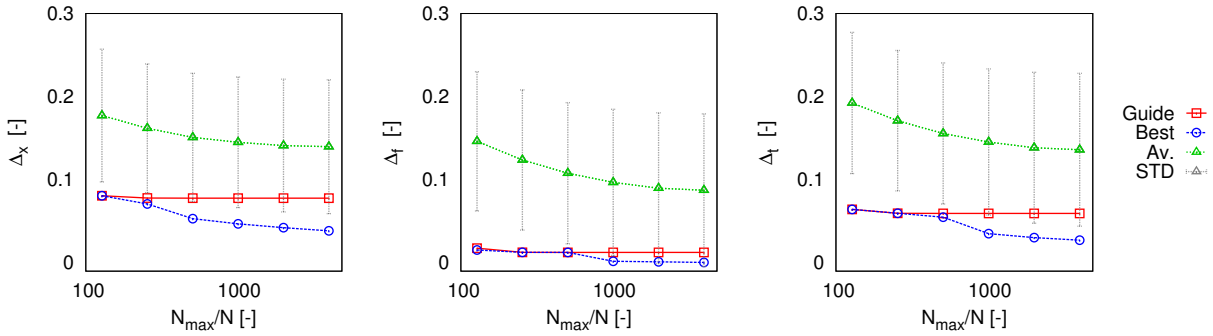


(a) Δ_x (b) Δ_f (c) Δ_t

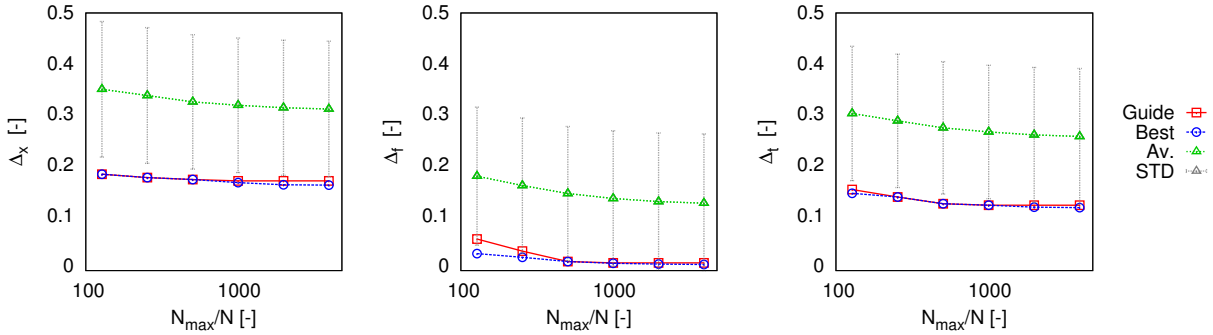
Figure 5.8: Relative variance $\sigma_r^2(\%)$ of A-DPSO setting parameters for $N \geq 10$

Table 5.4: Best performing setups for A-DPSO, $N \geq 10$

N_{max}/N	Average (STD)			Best A-DPSO													
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N}$	Init	Coef	Wall	Δ_t
				4	A.1	4	SEW	0.180	2	A.1	1	IW	0.026	4	A.1	4	SEW	0.136
	0.345	0.172	0.297	8	C.1	4	SEW	0.183	4	A.1	3	SEW	0.027	4	C.1	3	SEW	0.144
128				4	C.1	3	SEW	0.187	2	C.1	3	SEW	0.028	2	A.1	1	IW	0.146
	(0.132)	(0.132)	(0.130)	2	A.1	4	SEW	0.190	4	A.1	3	IW	0.030	4	A.1	4	IW	0.146
				8	A.1	3	SEW	0.191	4	A.1	4	SEW	0.032	2	A.1	4	SEW	0.147
				16	A.1	4	SEW	0.172	8	A.1	4	IW	0.018	8	A.1	4	IW	0.127
	0.331	0.152	0.280	8	A.1	4	IW	0.176	16	A.1	3	SEW	0.019	4	A.1	4	SEW	0.127
256				4	A.1	4	SEW	0.176	8	A.1	4	SEW	0.019	8	A.1	4	SEW	0.135
	(0.131)	(0.127)	(0.125)	32	A.1	3	SEW	0.179	4	A.1	4	SEW	0.019	8	C.1	4	SEW	0.138
				16	A.1	4	IW	0.181	8	C.1	3	SEW	0.023	8	A.1	3	SEW	0.138
				16	A.1	4	SEW	0.165	16	A.1	3	SEW	0.010	16	A.1	4	SEW	0.120
	0.316	0.135	0.265	16	A.1	4	IW	0.166	8	A.1	4	IW	0.014	16	A.1	4	IW	0.121
512				32	A.1	3	SEW	0.168	8	A.1	4	SEW	0.014	8	A.1	4	IW	0.123
	(0.128)	(0.124)	(0.125)	8	A.1	4	IW	0.172	16	A.1	3	IW	0.015	4	A.1	4	SEW	0.127
				8	C.1	4	SEW	0.175	16	A.1	4	IW	0.018	8	C.1	4	SEW	0.127
				16	A.1	4	SEW	0.157	32	C.1	3	SEW	0.006	16	A.1	4	SEW	0.112
	0.309	0.124	0.256	16	A.1	4	IW	0.162	16	A.1	3	SEW	0.007	16	A.1	4	IW	0.116
1024				32	A.1	3	SEW	0.163	32	A.1	3	SEW	0.011	32	A.1	3	SEW	0.118
	(0.127)	(0.125)	(0.125)	32	A.1	4	SEW	0.163	64	A.1	3	SEW	0.012	8	A.1	4	IW	0.123
				32	A.0	4	SEW	0.166	16	A.1	4	IW	0.012	16	C.1	4	SEW	0.123
				32	A.0	4	SEW	0.155	64	A.1	3	SEW	0.002	16	A.1	4	SEW	0.112
	0.304	0.118	0.250	16	A.1	4	SEW	0.156	32	C.1	3	SEW	0.005	32	A.1	4	SEW	0.113
2048				32	A.1	4	SEW	0.156	64	C.1	3	SEW	0.006	64	A.1	3	SEW	0.114
	(0.128)	(0.127)	(0.127)	16	A.1	2	IW	0.157	16	A.1	3	SEW	0.007	16	A.1	2	IW	0.114
				64	A.1	3	SEW	0.160	32	A.1	3	SEW	0.008	16	A.1	4	IW	0.115
				32	A.0	4	SEW	0.154	64	A.1	3	SEW	0.001	32	A.1	4	SEW	0.111
	0.301	0.114	0.246	32	A.1	4	SEW	0.155	32	C.1	3	SEW	0.005	16	A.1	4	SEW	0.111
4096				16	A.1	4	SEW	0.156	64	C.1	3	SEW	0.005	64	A.1	3	SEW	0.112
	(0.128)	(0.128)	(0.127)	16	A.1	2	IW	0.157	64	A.0	4	SEW	0.006	16	A.1	2	IW	0.113
				64	A.1	3	SEW	0.159	16	A.1	3	SEW	0.007	16	A.1	4	IW	0.115
				16	A.1	4	SEW	0.167	8	A.1	4	SEW	0.018	16	A.1	4	SEW	0.127
	0.317	0.136	0.265	16	A.1	4	IW	0.172	8	A.1	4	IW	0.018	8	A.1	4	IW	0.128
Av.				32	A.1	3	SEW	0.173	16	A.1	3	SEW	0.020	4	A.1	4	SEW	0.129
	(0.128)	(0.125)	(0.125)	8	A.1	4	IW	0.176	4	A.1	4	SEW	0.021	16	A.1	4	IW	0.131
				8	C.1	4	SEW	0.177	8	C.1	3	SEW	0.023	8	C.1	4	SEW	.132

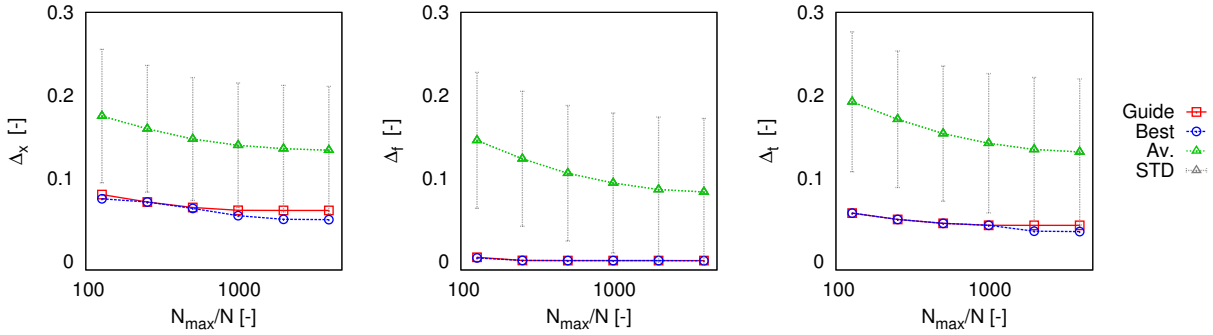


(a) $N < 10$

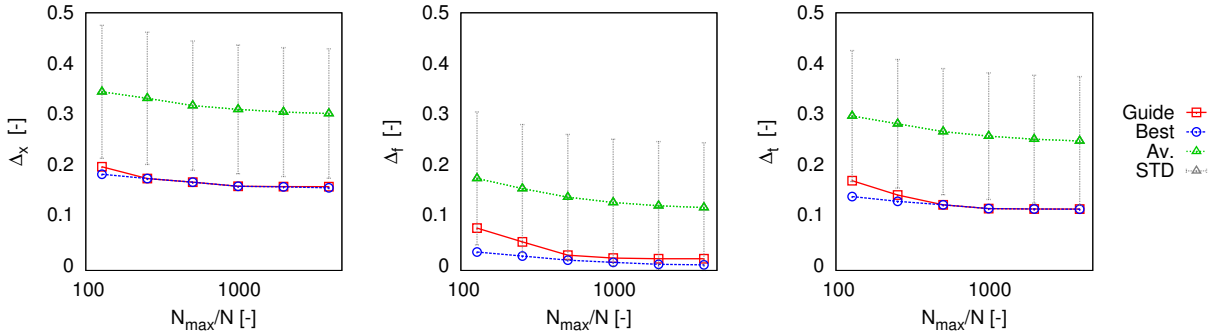


(b) $N \geq 10$

Figure 5.9: Performance of suggested guidelines using S-DPSO



(a) $N < 10$



(b) $N \geq 10$

Figure 5.10: Performance of suggested guidelines using A-DPSO

Table 5.5: Suggested guideline for S-DPSO and A-DPSO

	N	N_p/N	Initialization*	Coefficient set§	Wall-type
S-DPSO	< 10	4	C.1	4	SEW
	≥ 10	16	A.1	3	SEW
A-DPSO	< 10	4	C.1	4	IW/SEW
	≥ 10	16	A.1	4	SEW

* See Tab. 4.2
§ See Tab. 4.3

5.1.2 LS-DF_PSO

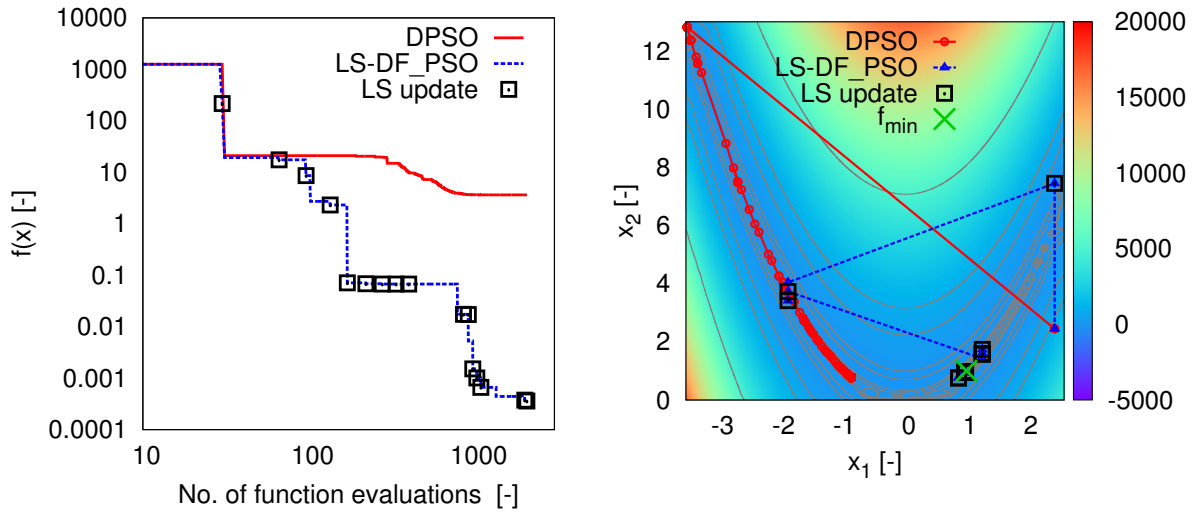
Numerical results are presented for the Rosenbrock function. The deterministic implementation of DPSO suggested as guideline in Sec. 5.1.1 and presented in [44] is used for the extension to the hybrid LS-DF_PSO. Specifically, a number of particles equal to $4N$ is used, with initialization over the variables domain by Hammersely sequence sampling, and DPSO coefficients given by Clerc [110], i.e., $\chi = 0.721$, $c_1 = c_2 = 1.655$.

Specifically, the minimization in $\mathbb{R}^2$ of the Rosenbrock function,

$$f(\mathbf{x}) = (a - x_1)^2 + b(x_2 - x_1^2)^2 \tag{5.1}$$

with $a = 1$, $b = 100$, $-20 \leq \mathbf{x} \leq 20$, is used as an explanatory test problem.

Figure 5.11 (a) shows the convergence of the LS-DF_PSO algorithm, compared to the standard DPSO. Black squares indicates LS-DF_PSO iterations where the line-search (LS) is used to improve the optimum location. Figure 5.11 (b) shows a comparison of the algorithms' convergence in a close up of the variables domain. The global-optimum location history is depicted, along with the real minimum ($f_{\min}$), which is located at $\mathbf{x} = \{1, 1\}$. The beneficial effects of using DPSO with LS are evident, providing a faster and more effective convergence to the optimum, along with the identification of the proper region of the global optimum.

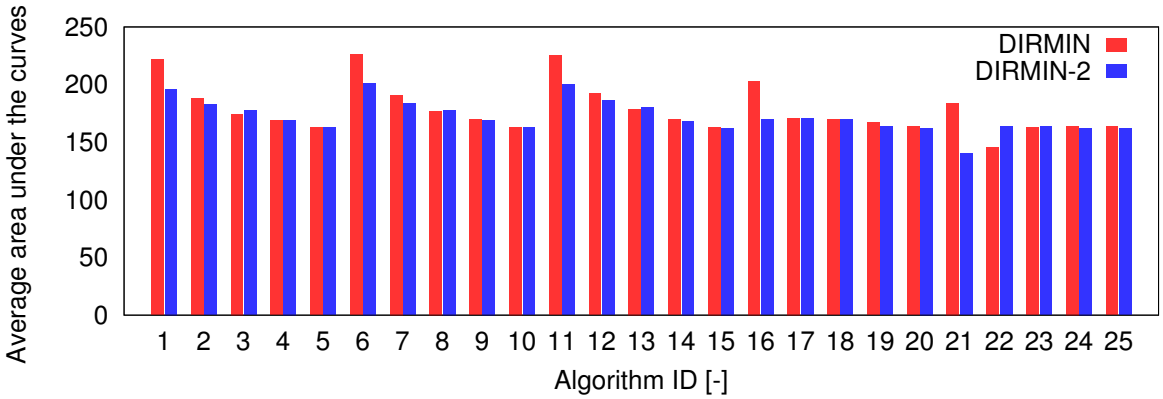


(a) (b)

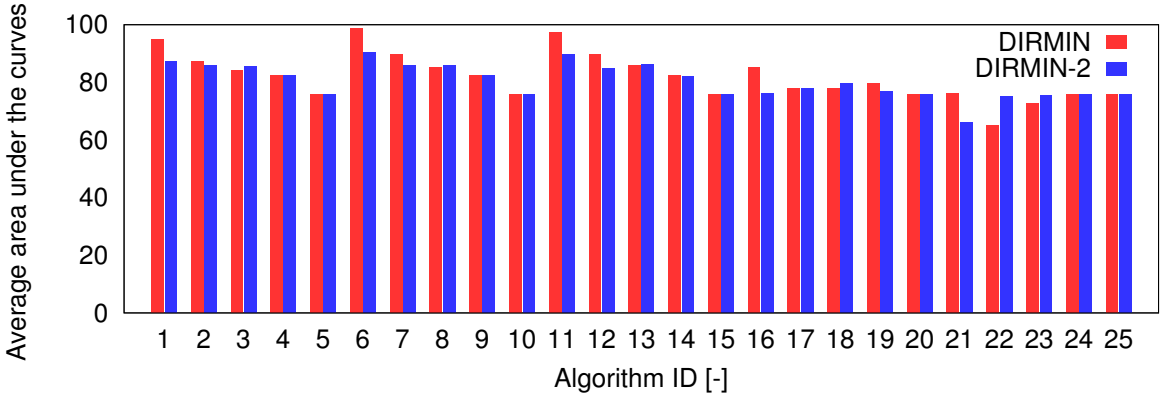
Figure 5.11: LS-DF_PSO convergence (a) and global-optimum location history (b) for the Rosenbrock function

5.1.3 DIRMIN and DIRMIN-2

Figure 5.12 shows the areas under (a) data and (b) performance profile curves of DIRMIN and DIRMIN-2, for the test problem with $N \leq 6$. Note that the higher bar correspond to the better algorithm. The most promising algorithms' setup, for both DIRMIN and DIRMIN-2, appears to be setup 6, corresponding to $\beta = 10^{-4}$ and $\gamma = 0$. Figures 5.13a and 5.13b, show the performances of DIRMIN and DIRMIN-2 in terms of the absolute metrics Δ_x, Δ_f and Δ_t, conditional to β and γ, respectively. These results suggest starting the derivative-free local minimizations from the very beginning ($\gamma = 0$) of the optimization procedure. Furthermore, for small problems ($N \leq 6$) a quite strict tolerance ($\beta = 10^{-4}$) for each derivative-free local minimization is advisable. Figure 5.13c shows the relative variability σ_r^2 for Δ_x, Δ_f and Δ_t respectively, associated to β and γ; γ is found the most significant parameter for both DIRMIN and DIRMIN-2. The performances of the whole set of DIRMIN and DIRMIN-2's setups, in terms of Δ_x, Δ_f and Δ_t are shown in Fig. 5.13d and 5.13e (note that the lower bar correspond to the better algorithm), respectively. For current cases, the most promising setups based on the absolute metrics are DIRMIN(21) (i.e., $\beta = 10^{-1}$, $\gamma = 0$) and DIRMIN-2(12) (i.e., $\beta = 10^{-3}$, $\gamma = 0.25$).

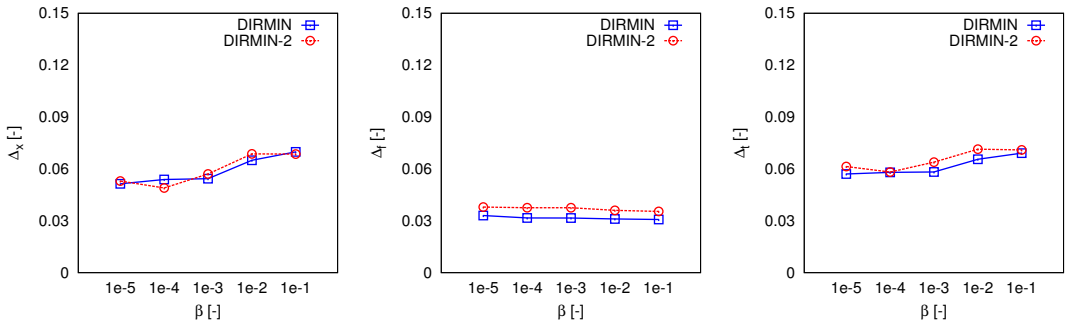


(a) Areas under the data profiles

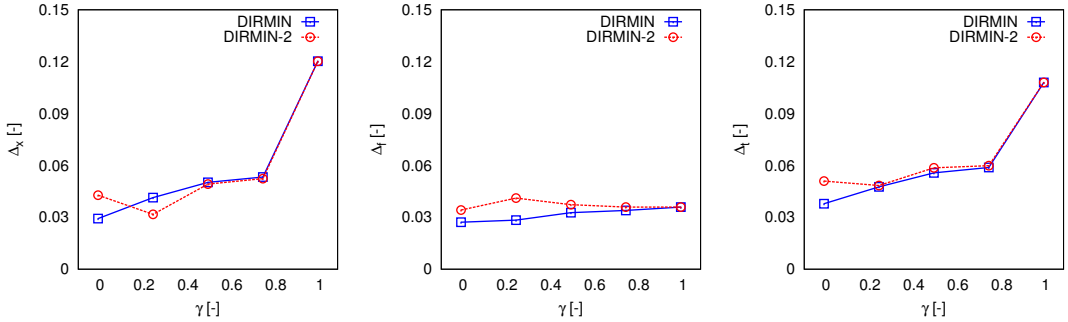


(b) Areas under the performance profiles

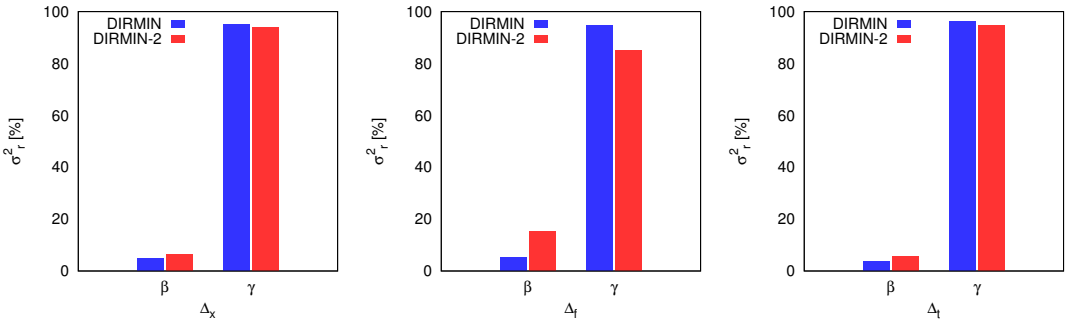
Figure 5.12: DIRMIN and DIRMIN-2 areas under the data and performance profiles for $N \leq 6$



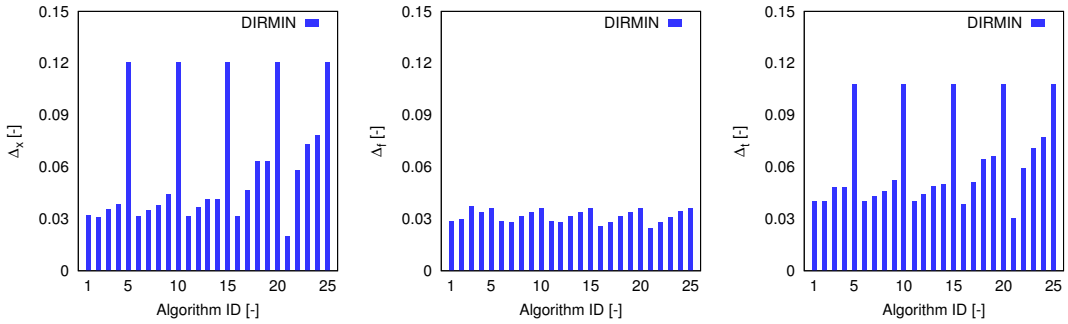
(a) Average performance, conditional to the tolerance β



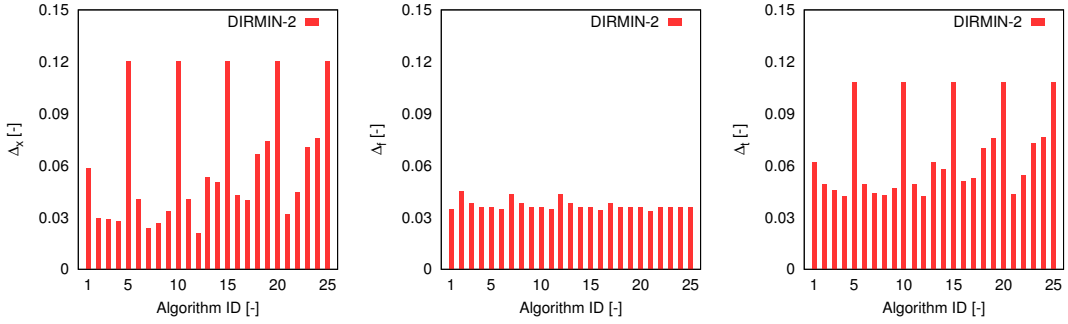
(b) Average performance, conditional to the activation trigger γ



(c) Relative variability of performance σ_r^2



(d) Average performance, conditional to DIRMIN setup



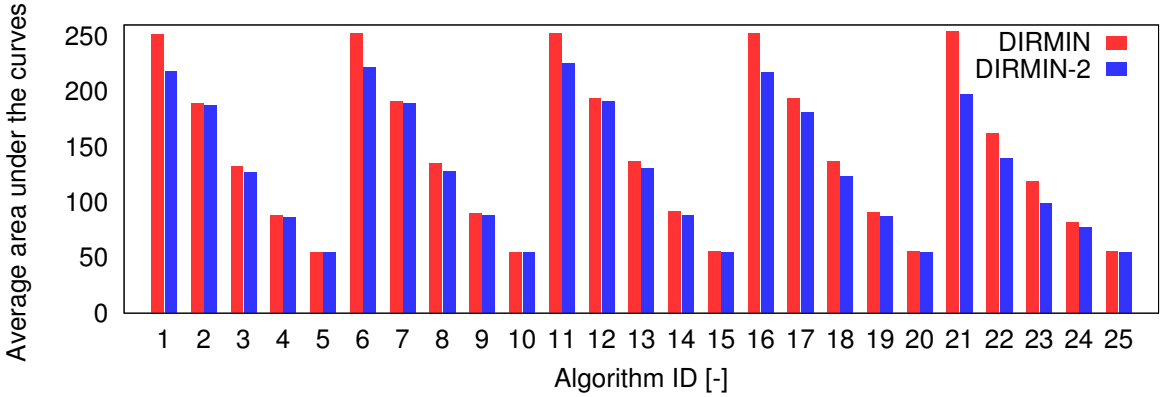
(e) Average performance, conditional to DIRMIN-2 setup

Figure 5.13: DIRMIN and DIRMIN-2 performance for test problem with $N \leq 6$

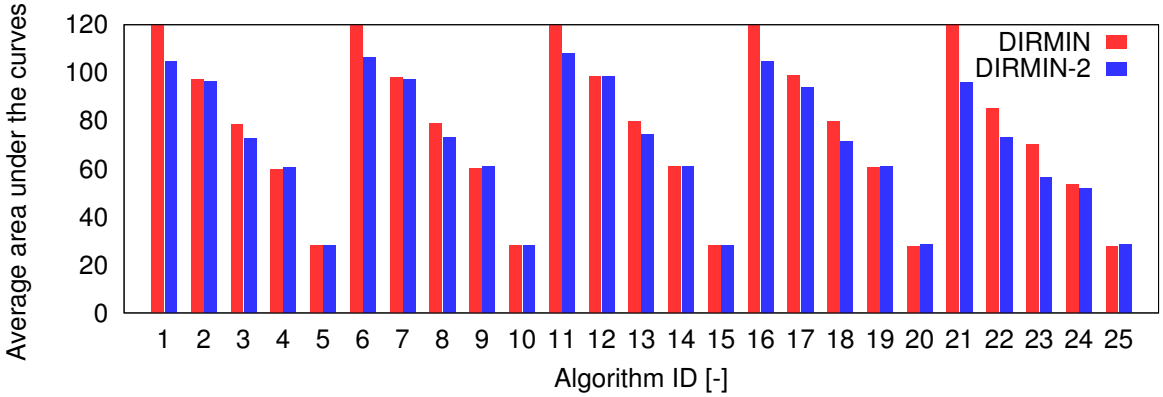
Figure 5.14 shows the areas under (a) data and (b) performance profile curves of DIRMIN and DIRMIN-2, for the test problem with $N > 6$. The most promising algorithms' setups appear to be number 21 (i.e., $\beta = 10^{-1}$ and $\gamma = 0$) and 11 (i.e., $\beta = 10^{-3}$ and $\gamma = 0$) for DIRMIN and DIRMIN-2, respectively. Figures 5.15a and 5.15b, show the performance of DIRMIN and DIRMIN-2 in terms of the absolute metrics Δ_x, Δ_f and Δ_t, conditional to β and γ, respectively. In this case, similarly to the small problems, it is beneficial to start the local minimizations at the very beginning of the optimization procedure, whereas higher β values are more advisable, even if the relative variability (see Fig. 5.15c) associated with β is almost equal to zero. Also for large problems ($N > 6$), γ is found the most significant parameter. Finally, the performances of the whole set of DIRMIN and DIRMIN-2's setups, in terms of Δ_x, Δ_f and Δ_t are shown in Fig. 5.15d and 5.15e, respectively. For current cases, the most promising setups based on the absolute metrics are DIRMIN(21) (i.e., $\beta = 10^{-1}$ and $\gamma = 0$) and DIRMIN-2(16) (i.e., $\beta = 10^{-2}$ and $\gamma = 0$).

The rationale behind the difference between the βs' results could be that when $N \leq 6$ some function evaluations can be devoted to achieving high precisions. This is not the case for large problems especially when considering that the adopted derivative-free local minimizations perform a sampling along the coordinate directions thus being potentially very costly when the number of variables is large.

Finally, Tab. 5.6 summarized the DIRMIN and DIRMIN-2 suggested setup for problem with less and more than six variables.



(a) Areas under the data profiles

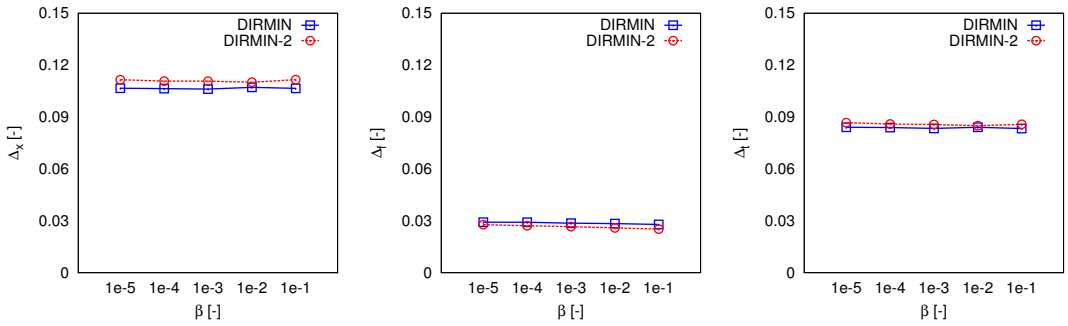


(b) Areas under the performance profiles

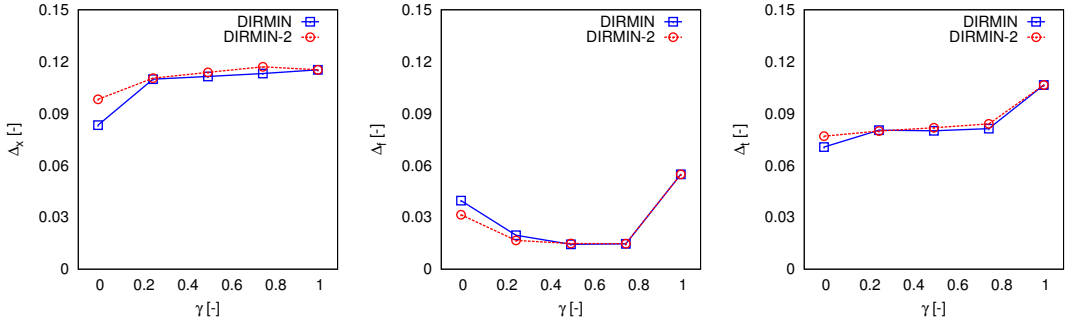
Figure 5.14: DIRMIN and DIRMIN-2 areas under the data and performance profiles for $N > 6$

Table 5.6: DIRMIN and DIRMIN-2 suggested setup

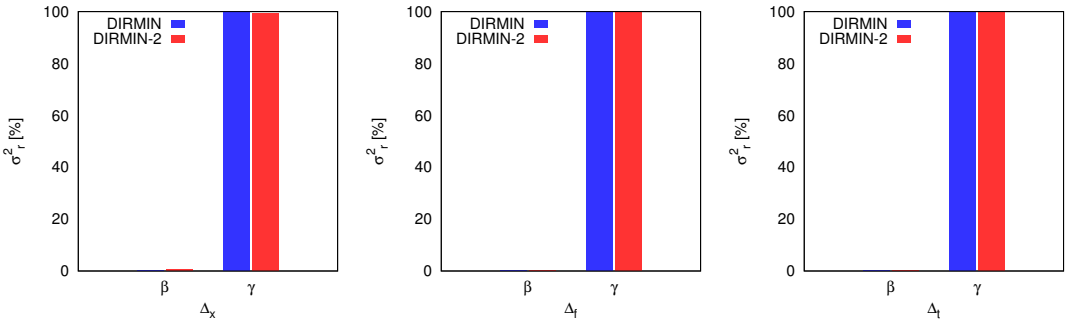
N	Algorithm	β	γ
≤ 6	DIRMIN	0.100	0.000
	DRIMIN-2	0.001	0.250
> 6	DIRMIN	0.100	0.000
	DRIMIN-2	0.010	0.000



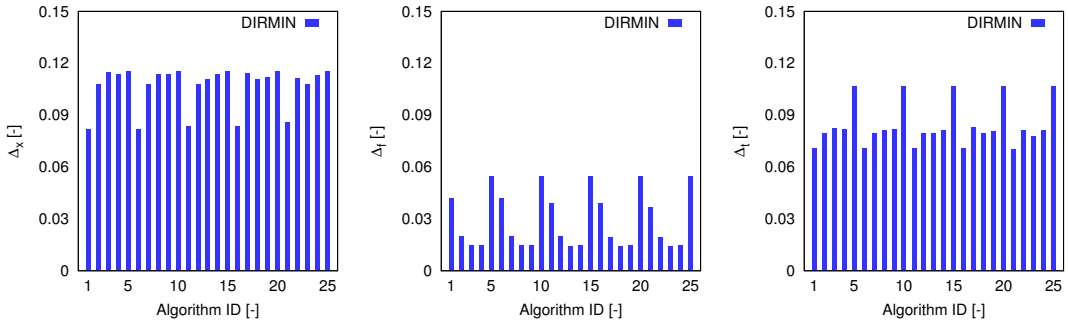
(a) Average performance, conditional to the tolerance β



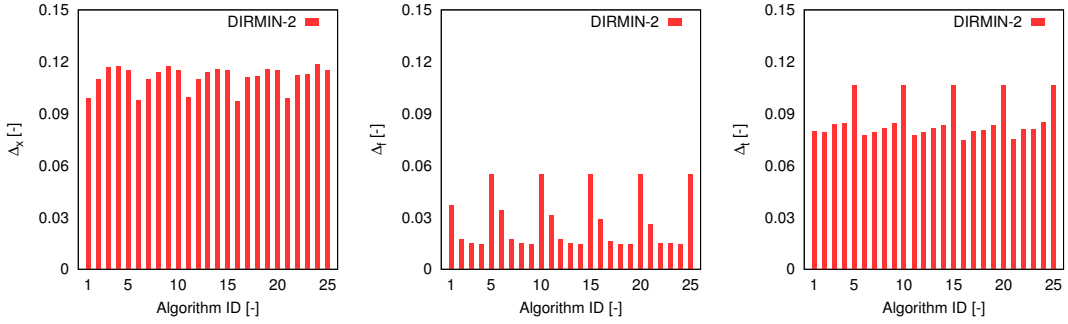
(b) Average performance, conditional to the activation trigger γ



(c) Relative variability of performance σ_r^2



(d) Average performance, conditional to DIRMIN setup



(e) Average performance, conditional to DIRMIN-2 setup

Figure 5.15: DIRMIN and DIRMIN-2 performance for test problem with $N > 6$

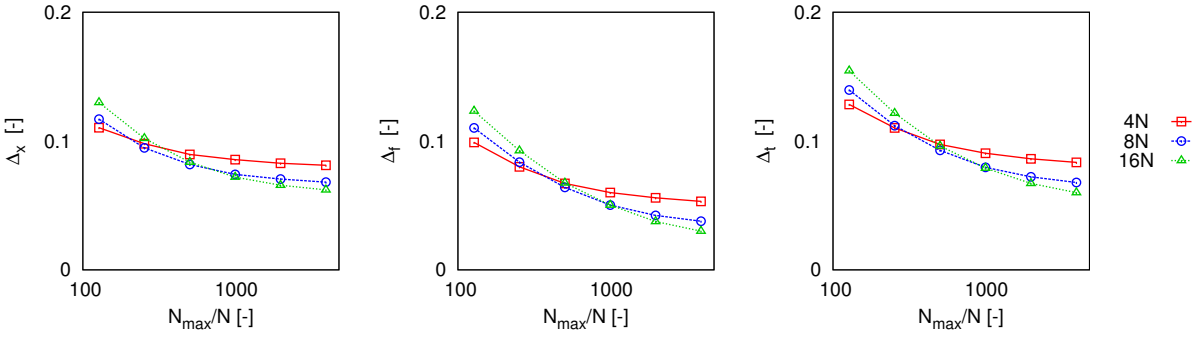
5.1.4 FSA

Figure 5.16 shows the performances of FSA in terms of the absolute metrics Δ_x, Δ_f, and Δ_t, for $N < 10$. Average values are presented, conditional to number of individuals (Fig. 5.16a), shoal initialization (Fig. 5.16b), coefficients ξ (Fig. 5.16c), q (Fig. 5.16d), and p (Fig. 5.16e), and wall-type approach (Fig. 5.16f), with $\alpha = 0.5$. Figure 5.17 shows the relative variability σ_r^2 for Δ_x, Δ_f, and Δ_t, respectively, associated to the FSA tuning parameters; the coefficient q (weight of the attraction forces) and the shoal initialization are the most significant parameters for low budget of function evaluations, whereas all the tuning parameters tend to have a similar importance with the increasing of the budget of function evaluations. Table 5.7 shows the five best performing setups for each Δ_x, Δ_f, and Δ_t, varying the budget of function evaluations. Average values and standard deviation for all FSA setups are also provided.

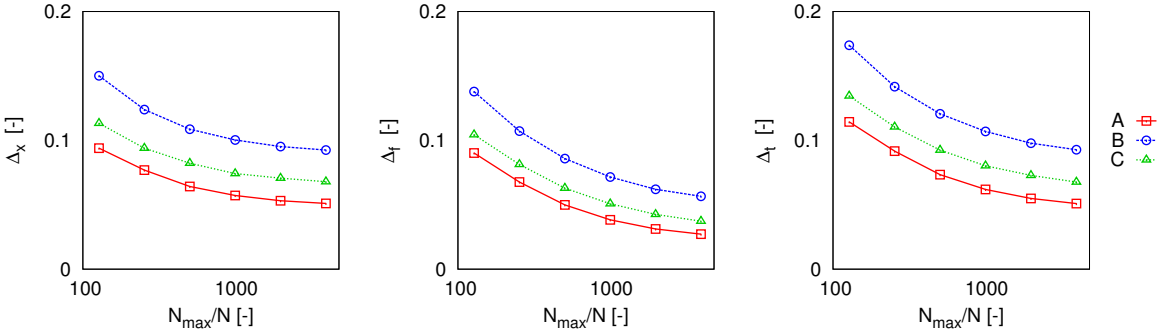
Similarly, the performances of FSA for the test functions with $N \geq 10$, are shown in Figs. 5.18 and 5.19, and summarized in Tab. 5.8. In particular, looking at the relative variability (Fig. 5.19), it can be seen how the shoal initialization become the most significant parameters for problem with a greater number of variables, for each budget of function evaluations.

The average value, with respect to the budget of function evaluations (Tab. 5.7 and 5.8), are used to define a reasonable guideline for the use of FSA in SBD optimization. The suggested guideline is summarized in Tab. 5.9. Figures 5.20a and 5.20b show the performances of the suggested setup, for $N < 10$ and $N \geq 10$, respectively. Average performance, standard deviation (STD) and best performing setup among all combinations are also shown for each budget. The guideline setup ("Guide") are found close or coincident to the "Best".

Finally, Figs. 5.21a and 5.21b show a comparison between the suggested FSA setups, DPSO [44], and DIRECT [45] for $N < 10$ and $N \geq 10$, respectively; FSA outperform DPSO and DIRECT, specially in terms of Δ_x for $N < 10$.

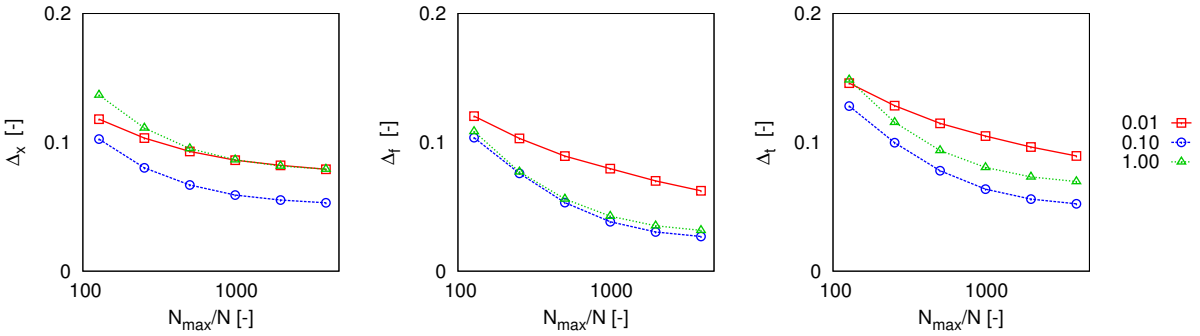


(a) Average performance, conditional to shoal size N_s

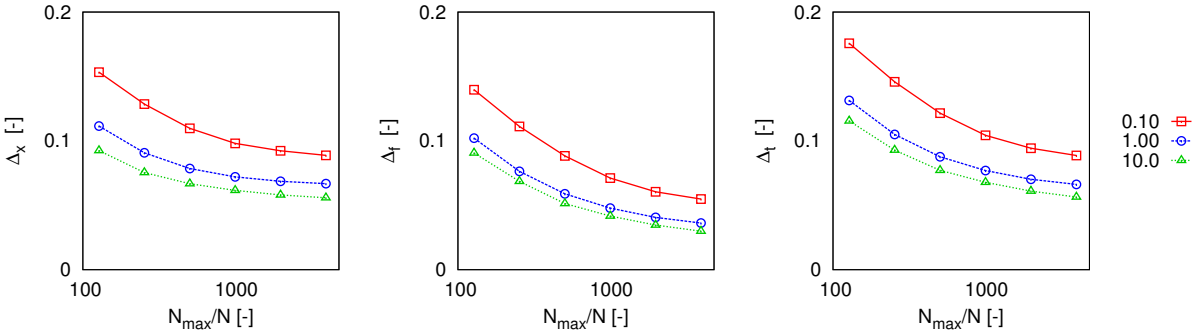


(b) Average performance, conditional to shoal initialization

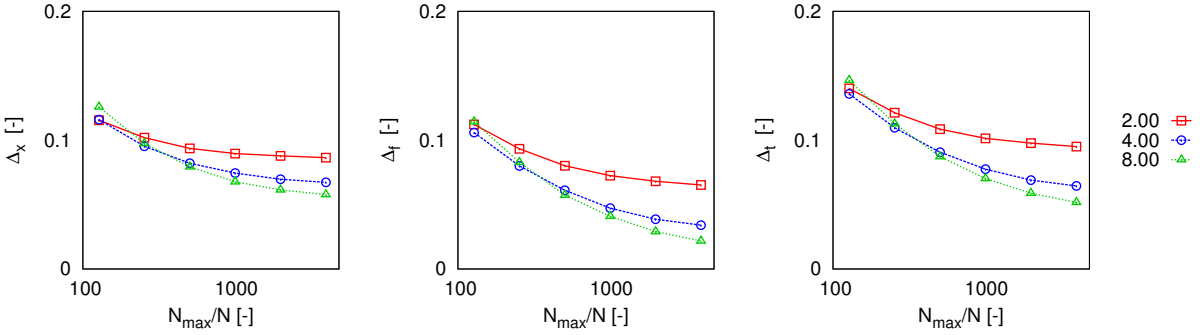
Figure 5.16: FSA average performance for $N < 10$



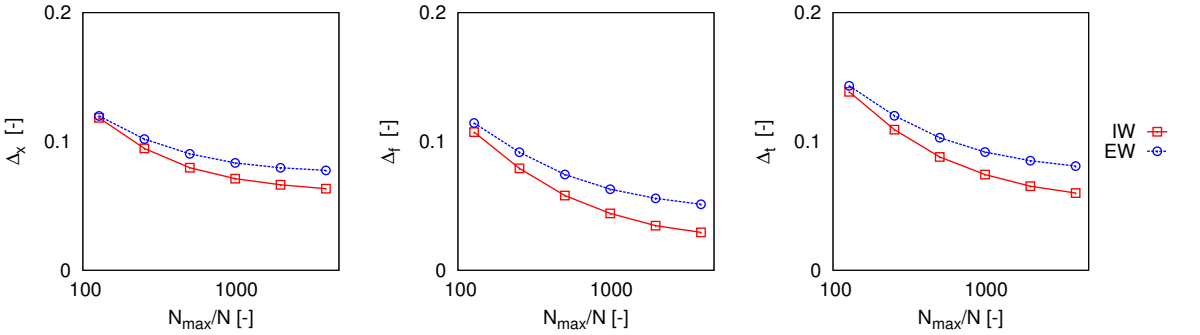
(c) Average performance, conditional to coefficient ξ



(d) Average performance, conditional to coefficient q

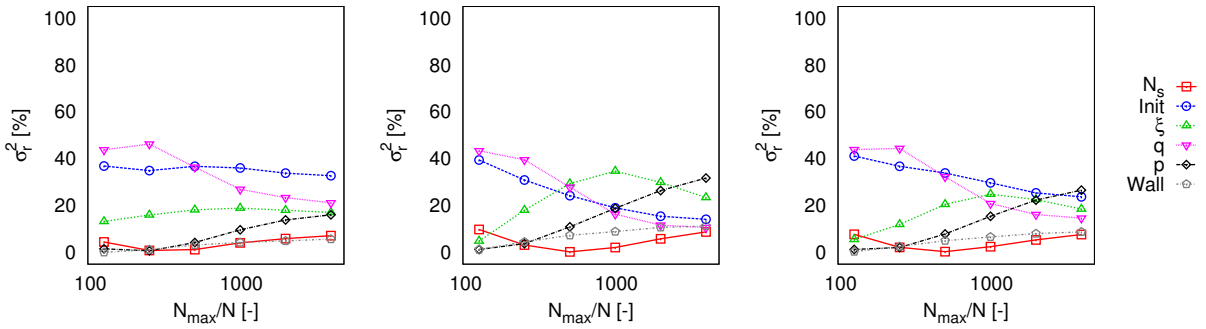


(e) Average performance, conditional to coefficient p



(f) Average performance, conditional to wall-type approach

Figure 5.16 (cont'd): FSA average performance for $N < 10$

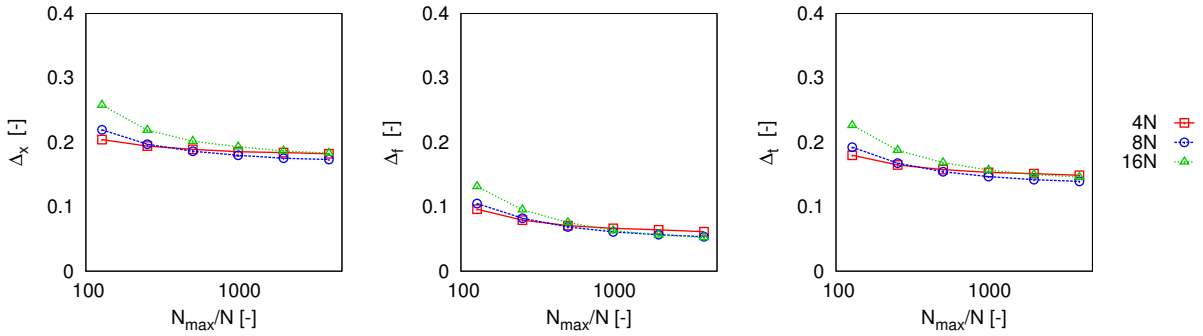


(a) Δ_x (b) Δ_f (c) Δ_t

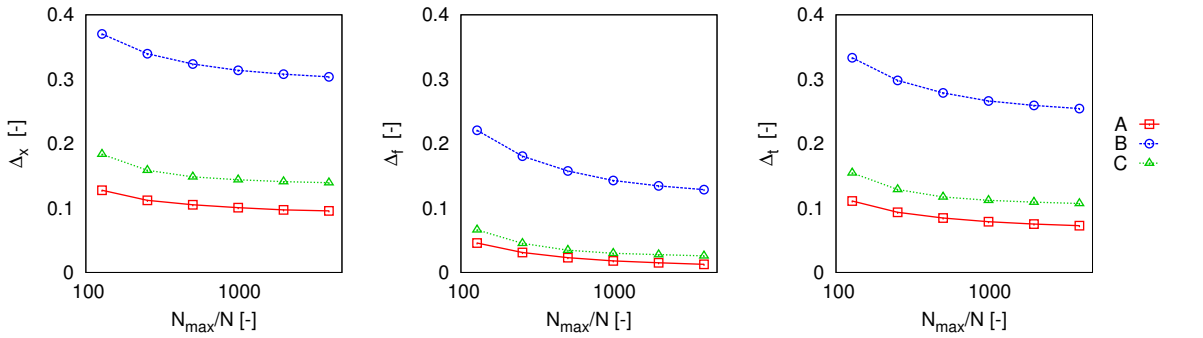
Figure 5.17: Relative variance $\sigma_r^2(\%)$ of FSA for $N < 10$

Table 5.7: Best performing setups for FSA, $N < 10$

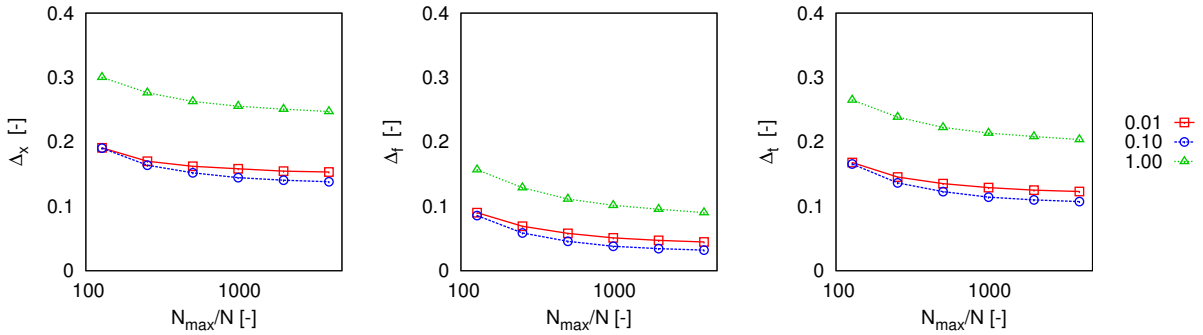
$N_{\max}/N$	Average (STD)			Best FSA													
	Δ_x	Δ_f	Δ_t	$\frac{N_s}{N}$	Init	ξ,q,p	Wall	Δ_x	$\frac{N_s}{N}$	Init	ξ,q,p	Wall	Δ_f	$\frac{N_s}{N}$	Init	ξ,q,p	Wall	Δ_t
				16	A	3,3,1	IW	0.346e-1	8	A	3,2,2	EW	0.180e-1	16	A	3,3,1	IW	0.448e-1
	1.191e-1	1.108e-1	1.409e-1	4	A	2,1,2	IW	0.451e-1	16	A	3,2,1	EW	0.201e-1	16	A	3,3,1	EW	0.507e-1
128				16	A	3,3,1	EW	0.464e-1	16	A	3,2,1	IW	0.208e-1	8	A	3,3,2	IW	0.532e-1
	(5.404e-2)	(4.733e-2)	(5.487e-2)	4	C	2,2,2	EW	0.471e-1	4	C	3,2,2	EW	0.250e-1	4	A	3,2,2	EW	0.551e-1
				8	A	3,3,2	IW	0.477e-1	4	A	3,2,2	EW	0.258e-1	16	A	3,2,1	EW	0.552e-1
				16	A	3,3,2	EW	0.324e-1	4	A	3,2,3	EW	0.165e-2	4	A	3,2,3	EW	0.259e-1
	9.823e-2	8.537e-2	1.146e-2	16	A	2,3,3	EW	0.344e-1	8	A	3,2,2	EW	0.246e-2	8	A	3,3,2	IW	0.333e-1
256				4	A	3,2,3	EW	0.356e-1	8	A	3,3,2	EW	0.371e-2	4	A	2,1,2	IW	0.336e-1
	(4.937e-2)	(4.744e-2)	(5.274e-2)	4	A	2,1,3	IW	0.371e-1	8	A	3,2,2	IW	0.575e-2	16	A	3,3,2	EW	0.345e-1
				16	A	3,3,2	IW	0.391e-1	8	A	3,3,2	IW	0.628e-2	16	A	3,3,1	IW	0.357e-1
				4	B	2,1,3	EW	0.243e-1	4	A	3,2,3	EW	0.706e-3	4	A	2,1,3	IW	0.190e-1
	8.501e-2	6.620e-2	9.545e-2	4	A	2,1,3	IW	0.265e-1	8	A	3,2,3	EW	0.832e-3	16	A	3,3,2	IW	0.203e-1
512				16	A	3,3,2	IW	0.275e-1	8	A	3,2,2	EW	0.969e-3	16	A	3,3,2	EW	0.211e-1
	(4.609e-2)	(4.735e-2)	(5.148e-2)	4	A	2,1,3	EW	0.277e-1	4	A	2,1,3	IW	0.123e-2	4	A	2,1,3	EW	0.214e-1
				16	A	3,3,2	EW	0.287e-1	8	A	3,3,3	EW	0.134e-2	4	B	2,1,3	EW	0.214e-1
				16	A	2,1,3	EW	0.198e-1	8	A	3,3,3	IW	0.406e-3	4	A	2,1,3	IW	0.172e-1
	7.724e-2	5.354e-2	8.303e-2	16	A	2,2,3	EW	0.209e-1	16	A	3,2,2	EW	0.480e-3	4	B	2,1,3	EW	0.176e-1
1024				4	B	2,1,3	EW	0.215e-1	8	A	3,2,3	EW	0.480e-3	4	A	2,1,3	EW	0.190e-1
	(4.434e-2)	(4.596e-2)	(5.001e-2)	16	A	2,2,3	IW	0.238e-1	8	A	3,3,3	EW	0.553e-3	16	A	3,3,2	IW	0.192e-1
				4	A	2,1,3	IW	0.241e-1	8	A	2,1,3	IW	0.606e-3	16	A	3,2,3	IW	0.194e-1
				16	A	2,2,3	IW	0.123e-1	16	A	3,2,2	EW	0.179e-3	16	A	2,1,3	EW	0.116e-1
	7.297e-2	4.524e-2	7.516e-2	16	A	2,1,3	EW	0.160e-1	16	A	3,2,3	IW	0.196e-3	8	C	2,2,3	EW	0.128e-1
2048				8	C	2,2,3	EW	0.165e-1	8	A	3,3,3	IW	0.378e-3	16	A	2,2,3	IW	0.129e-1
	(4.349e-2)	(4.511e-2)	(4.928e-2)	16	A	2,2,3	EW	0.181e-1	16	A	2,1,3	IW	0.444e-3	4	B	2,1,3	EW	0.161e-1
				4	B	2,1,3	EW	0.214e-1	8	A	3,2,3	EW	0.466e-3	4	A	2,1,3	IW	0.171e-1
				8	C	2,2,3	EW	0.810e-2	16	A	3,2,2	EW	0.160e-3	8	C	2,2,3	EW	0.647e-2
	7.046e-2	4.033e-2	7.041e-2	16	A	2,2,3	EW	0.983e-2	16	A	3,2,3	IW	0.170e-3	16	A	2,2,3	EW	0.722e-2
4096				16	A	2,2,3	IW	0.102e-1	16	A	3,2,3	EW	0.314e-3	16	A	2,2,3	IW	0.778e-2
	(4.329e-2)	(4.499e-2)	(4.934e-2)	16	A	2,1,3	EW	0.154e-1	16	A	3,2,2	IW	0.325e-3	16	A	2,1,3	EW	0.111e-1
				16	A	1,2,2	IW	0.179e-1	8	A	3,3,3	IW	0.357e-3	4	B	2,1,3	EW	0.153e-1
				4	A	2,1,2	IW	0.344e-1	8	A	3,2,2	EW	0.390e-2	4	A	3,2,3	EW	0.302e-1
	8.716e-2	6.691e-2	9.659e-2	4	A	2,1,3	IW	0.348e-1	4	A	3,2,3	EW	0.510e-2	8	A	3,3,2	IW	0.310e-1
Av.				16	A	2,3,3	EW	0.352e-1	8	A	3,3,2	EW	0.638e-2	16	A	3,3,1	IW	0.321e-1
	(4.455e-2)	(4.310e-2)	(4.825e-2)	16	A	3,3,1	IW	0.354e-1	8	A	3,3,2	IW	0.749e-2	4	A	2,1,2	IW	0.321e-1
				4	B	2,1,3	EW	0.361e-1	16	A	3,2,1	EW	0.899e-2	4	A	2,1,3	IW	0.338e-1



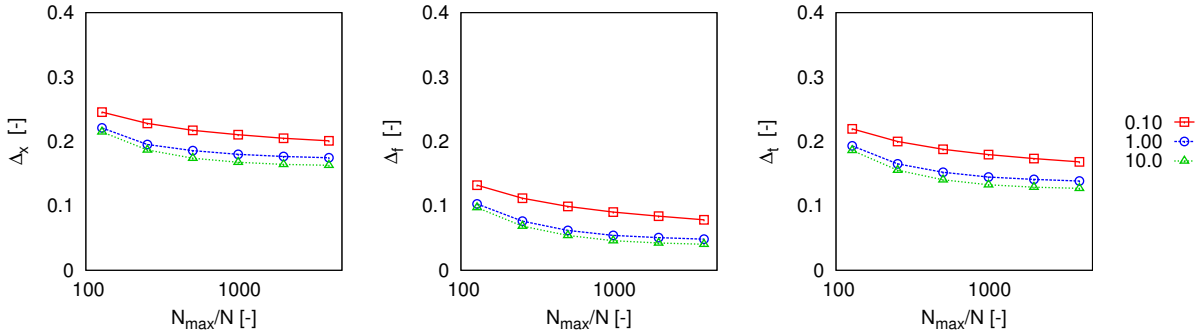
(a) Average performance, conditional to shoal size N_s



(b) Average performance, conditional to shoal initialization

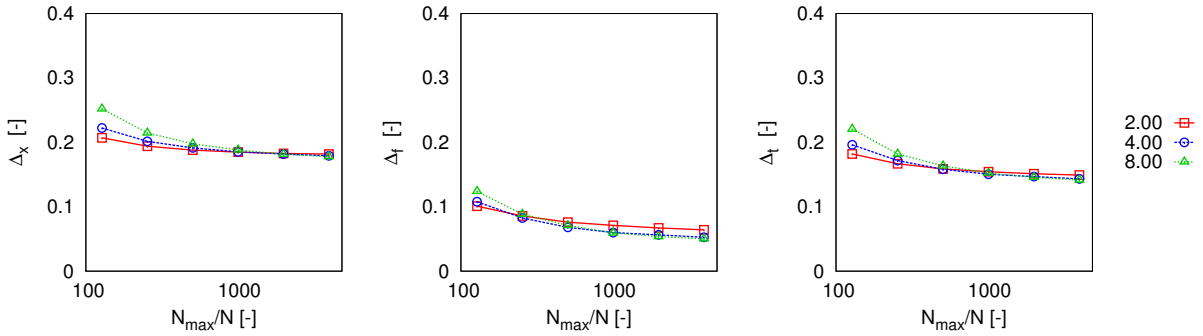


(c) Average performance, conditional to coefficient ξ

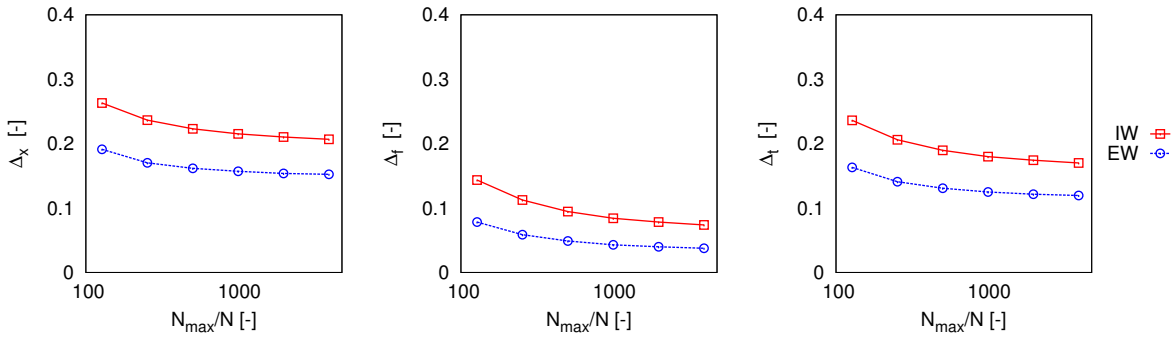


(d) Average performance, conditional to coefficient q

Figure 5.18: FSA average performance for $N \geq 10$

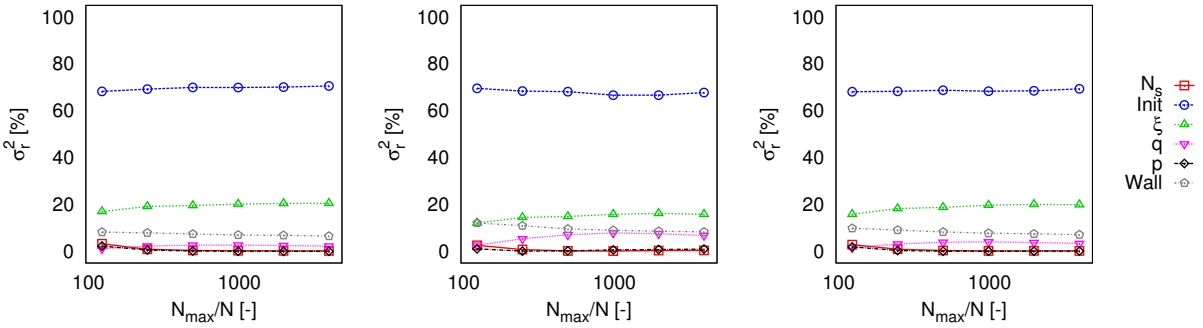


(e) Average performance, conditional to coefficient p



(f) Average performance, conditional to wall-type approach

Figure 5.18 (cont'd): FSA average performance for $N \geq 10$



(a) Δ_x (b) Δ_f (c) Δ_t

Figure 5.19: Relative variance $\sigma_r^2(\%)$ of FSA for $N \geq 10$

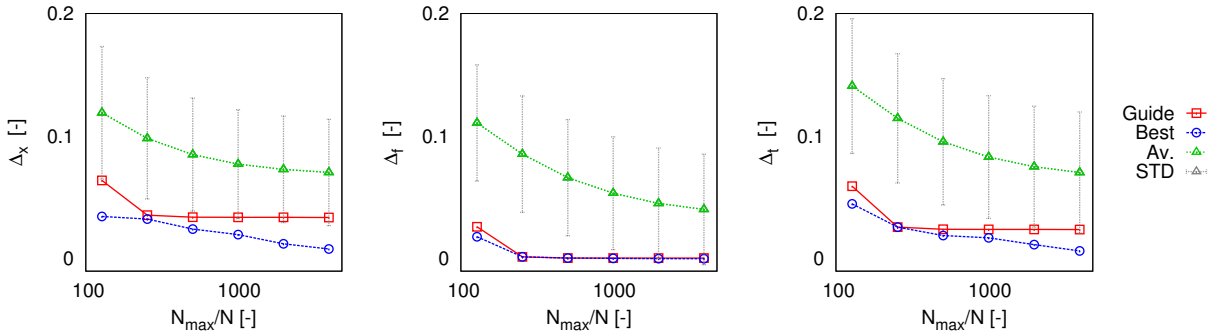
Table 5.8: Best performing setups for FSA, $N \geq 10$

$N_{\max}/N$	Average (STD)			Best FSA												
	Δ_x	Δ_f	Δ_t	$\frac{N_s}{N}$	Init	ξ,q,p	Wall	Δ_x	$\frac{N_s}{N}$	Init	ξ,q,p	Wall	Δ_f	$\frac{N_s}{N}$	Init	ξ,q,p	Wall	Δ_t
				4	B	2,1,2	EW	0.339e-1	4	A	3,3,2	IW	0.316e-2	4	B	2,1,2	EW	0.270e-1
	2.271e-1	1.109e-1	1.996e-1	8	A	2,1,3	EW	0.526e-1	4	A	3,3,2	EW	0.316e-2	8	A	2,1,3	EW	0.386e-1
128				16	A	2,1,3	EW	0.539e-1	4	A	2,1,1	EW	0.382e-2	16	A	2,1,3	EW	0.393e-1
	(1.466e-1)	(1.133e-1)	(1.367e-1)	4	A	2,1,3	EW	0.565e-1	16	A	2,1,3	EW	0.394e-2	4	A	2,1,3	EW	0.422e-1
				16	A	3,3,1	IW	0.625e-1	8	A	2,1,3	EW	0.482e-2	4	B	2,1,1	EW	0.453e-1
				4	B	2,1,2	EW	0.271e-1	4	A	2,1,2	EW	0.454e-3	4	B	2,1,2	EW	0.192e-1
	2.034e-1	8.556e-2	1.735e-1	4	B	2,1,3	EW	0.438e-1	4	A	3,3,2	EW	0.636e-3	4	B	2,1,3	EW	0.365e-1
256				8	B	2,1,2	EW	0.506e-1	4	A	3,3,2	IW	0.636e-3	8	A	2,1,3	EW	0.407e-1
	(1.383e-1)	(1.004e-1)	(1.278e-1)	16	C	1,1,3	EW	0.546e-1	4	A	2,1,2	IW	0.707e-3	8	B	2,1,2	EW	0.407e-1
				16	A	3,3,1	EW	0.555e-1	4	B	2,1,2	EW	0.114e-2	16	A	3,3,1	EW	0.428e-1
				4	B	2,1,2	EW	0.277e-1	8	A	2,1,2	EW	0.155e-3	4	B	2,1,2	EW	0.196e-1
	1.923e-1	7.169e-2	1.602e-2	8	B	2,1,3	EW	0.442e-1	4	C	2,1,2	EW	0.214e-3	16	A	2,1,3	EW	0.334e-1
512				16	A	2,1,3	EW	0.464e-1	4	A	2,1,3	EW	0.227e-3	4	B	2,1,3	EW	0.335e-1
	(1.329e-1)	(9.214e-2)	(1.216e-2)	4	B	2,1,3	EW	0.474e-1	4	A	2,1,2	EW	0.278e-3	4	C	2,1,3	EW	0.346e-1
				4	C	2,1,3	EW	0.488e-1	4	A	2,1,3	IW	0.292e-3	8	B	2,1,3	EW	0.347e-1
				4	B	2,1,2	EW	0.276e-1	8	A	2,1,2	EW	0.935e-4	4	B	2,1,2	EW	0.195e-1
	1.861e-1	6.348e-2	1.524e-1	8	B	2,1,3	EW	0.313e-1	4	C	2,1,3	EW	0.100e-3	8	B	2,1,3	EW	0.222e-1
1024				16	C	2,1,2	EW	0.430e-1	8	A	2,1,3	EW	0.116e-3	16	C	2,1,2	EW	0.305e-1
	(1.296e-1)	(8.676e-2)	(1.175e-1)	4	B	2,1,3	EW	0.434e-1	8	A	2,1,2	IW	0.127e-3	4	B	2,1,3	EW	0.307e-1
				4	C	2,1,3	EW	0.445e-1	4	A	2,1,3	EW	0.132e-3	4	C	2,1,3	EW	0.315e-1
				4	B	2,1,2	EW	0.275e-1	8	C	3,3,3	EW	0.412e-4	4	B	2,1,2	EW	0.194e-1
	1.820e-1	5.904e-2	1.478e-1	8	B	2,1,3	EW	0.280e-1	8	C	2,1,3	EW	0.562e-4	8	B	2,1,3	EW	0.198e-1
2048				16	C	2,1,2	EW	0.422e-1	16	A	2,1,2	EW	0.834e-4	16	C	2,1,2	EW	0.298e-1
	(1.278e-1)	(8.269e-2)	(1.151e-1)	4	B	2,1,3	EW	0.429e-1	4	C	2,2,3	EW	0.874e-4	4	B	2,1,3	EW	0.303e-1
				4	C	2,1,3	EW	0.429e-1	8	B	2,1,3	EW	0.888e-4	4	C	2,1,3	EW	0.303e-1
				4	B	2,1,2	EW	0.258e-1	16	C	3,3,3	EW	0.389e-4	4	B	2,1,2	EW	0.183e-1
	1.796e-1	5.567e-2	1.447e-1	8	B	2,1,3	EW	0.278e-1	8	C	3,3,3	EW	0.390e-4	8	B	2,1,3	EW	0.197e-1
4096				16	C	2,1,3	EW	0.419e-1	8	C	2,1,3	EW	0.537e-4	16	C	2,1,3	EW	0.296e-1
	(1.256e-1)	(7.954e-2)	1.126e-1	16	C	2,1,2	EW	0.421e-1	8	C	2,2,3	IW	0.561e-4	16	C	2,1,2	EW	0.298e-1
				4	B	2,1,3	EW	0.428e-1	16	B	2,1,3	EW	0.620e-4	4	B	2,1,3	EW	0.303e-1
				4	B	2,1,2	EW	0.283e-1	4	A	3,3,2	IW	0.834e-3	4	B	2,1,2	EW	0.205e-1
	1.951e-1	7.438e-2	1.630e-1	4	B	2,1,3	EW	0.535e-1	4	A	3,3,2	EW	0.834e-3	4	B	2,1,3	EW	0.395e-1
Av.				8	B	2,1,3	EW	0.564e-1	4	A	2,1,2	IW	0.116e-2	8	A	2,1,3	EW	0.410e-1
	(1.320e-1)	(9.109e-2)	1.206e-1	4	B	2,2,2	EW	0.568e-1	4	A	2,1,2	EW	0.132e-2	8	B	2,1,3	EW	0.417e-1
				8	A	2,1,3	EW	0.573e-1	16	A	2,1,3	EW	0.197e-2	16	A	2,1,3	EW	0.417e-1

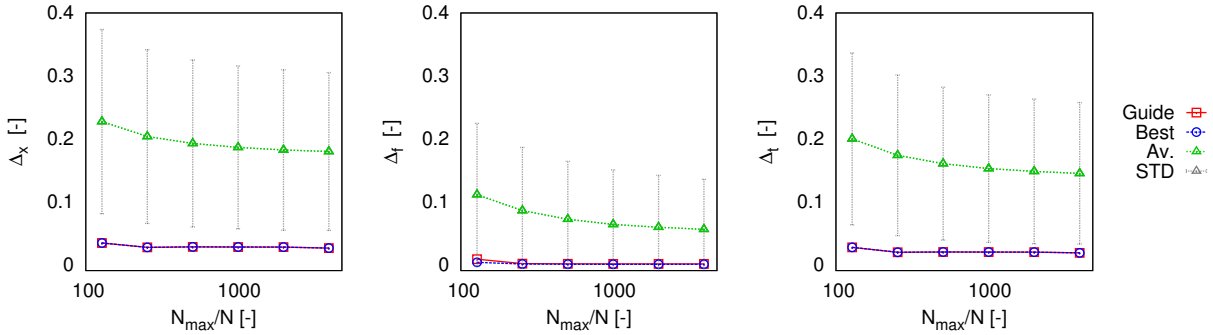
Table 5.9: Suggested guideline for FSA

N	N_s/N	Initialization*	α	ξ	q	p	Wall-type
< 10	4	A	0.50	1.00	1.00	8.00	EW
≥ 10	4	B	0.50	0.10	0.10	4.00	EW

* See Tab. 4.5

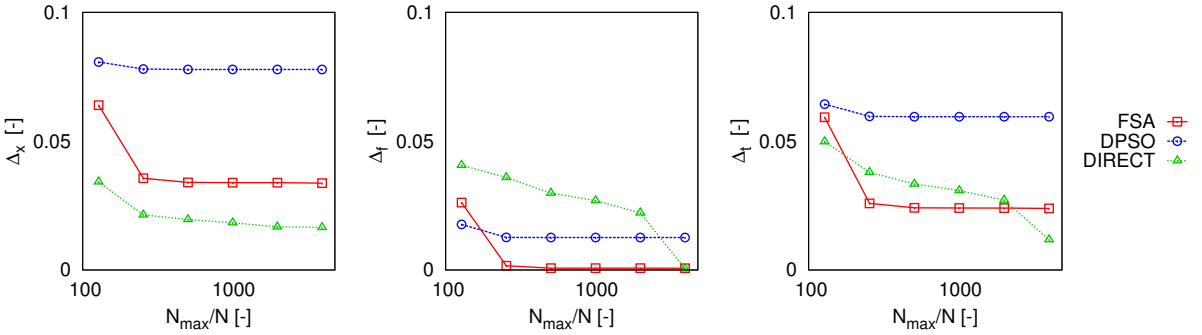


(a) $N < 10$

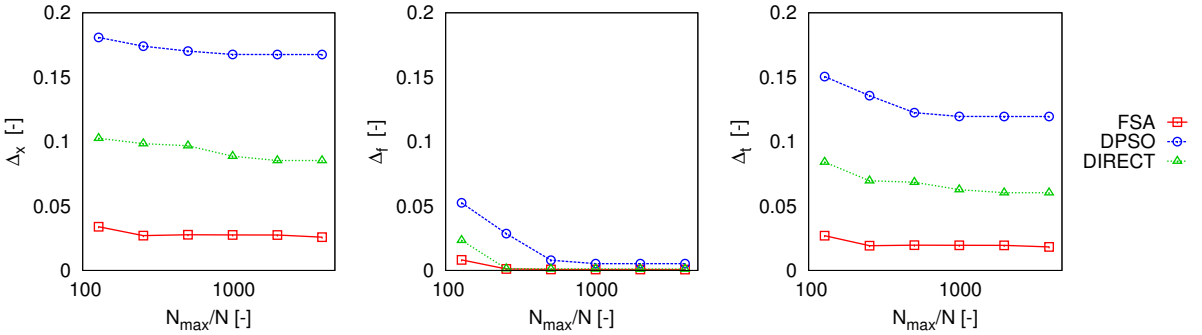


(b) $N \geq 10$

Figure 5.20: Performance of suggested guideline for FSA



(a) $N < 10$

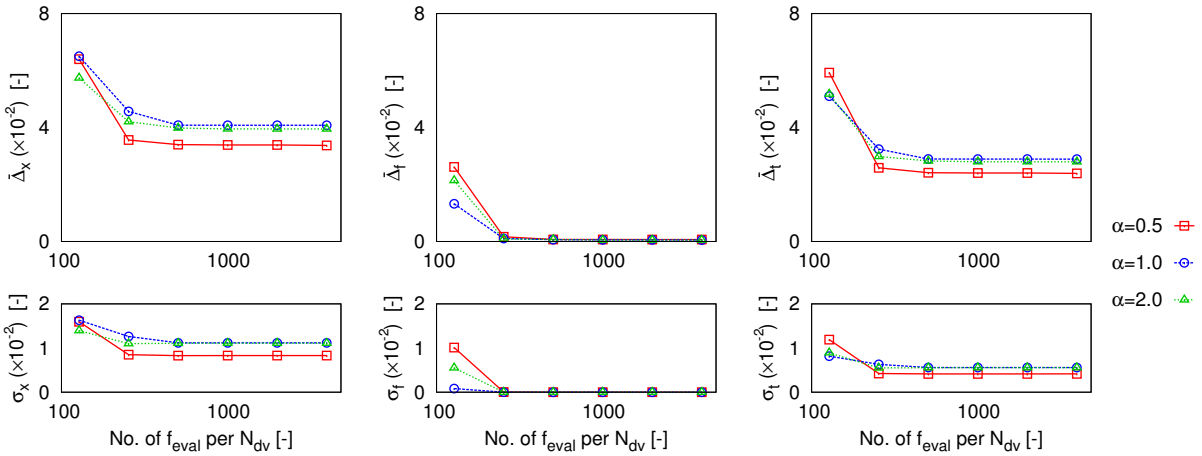


(b) $N \geq 10$

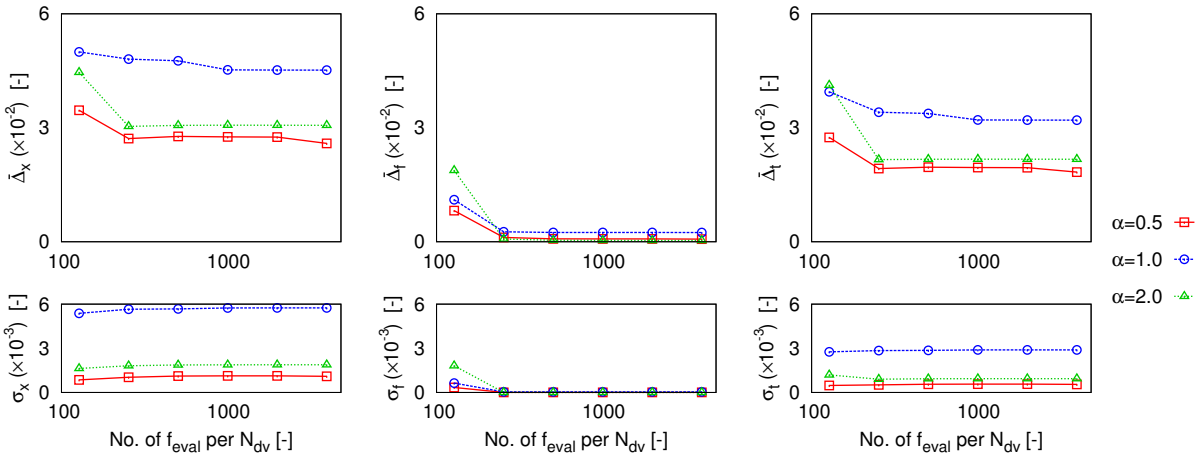
Figure 5.21: FSA suggested setup versus DPSO and DIRECT

An ex-post further study on the same 60 analytical test functions is used to identify the most promising food-related attraction force coefficient for FSA and its influence on the algorithm performances. Three values are used for the coefficient α, defining the food-related attraction force $\boldsymbol{\varphi}_j$ (Eq. 1.15): $\alpha = 0.5$, 1, and 2. The analysis are presented setting a part the results obtained for functions with less and more than 10 variables, as in previous subsection, using the guideline defined in Tab. 5.9.

Figure 5.22a and 5.22b shows the average values in terms of the absolute metrics $\bar{\Delta}_x$, $\bar{\Delta}_f$, and $\bar{\Delta}_t$, and the corresponding standard deviations σ_x, σ_f, and σ_t, conditional to α, for $N < 10$ and $N \geq 10$, respectively. The use of the coefficient $\alpha = 0.5$ shows the most effective and efficient results in term of algorithm convergence, for both test problems with less or more than 10 variables.



(a) $N < 10$



(b) $N \geq 10$

Figure 5.22: FSA food-related attraction force coefficients results for the analytical test functions

5.2 Simulation-based design problems

5.2.1 Delft catamaran

Problem I

A preliminary sensitivity analysis for each design variable is shown in Fig. 5.23, showing $\Delta f\%$ compared to the parent hull. Changes in f are found significant in each direction, revealing a reduction of the objective function close to 9%.

The optimization is performed with both S-DPSO and A-DPSO, as per the guideline suggested in Tab. 5.5. S-DPSO and A-DPSO iterations are shown in Fig. 5.24, revealing a quite sudden convergence. Optimization results, summarized in Tab. 5.10, show that S-DPSO and A-DPSO with both IW and SEW lead to a reduction of the objective function close to 20%. There are not significant differences between the results obtained by S-DPSO and A-DPSO, as shown in Figs. 5.25 and 5.26. The differences in optimal design variables are mainly due to IW or SEW. Furthermore the optimum configuration leads to a considerable reduction of wave's elevation compared to the original shape (Figs. 5.27).

Finally, it may be noted that the final configuration is fairly close (except for the second variable) to that obtained using metamodels and an URANS solver [47], used here for reference (Fig. 5.25).

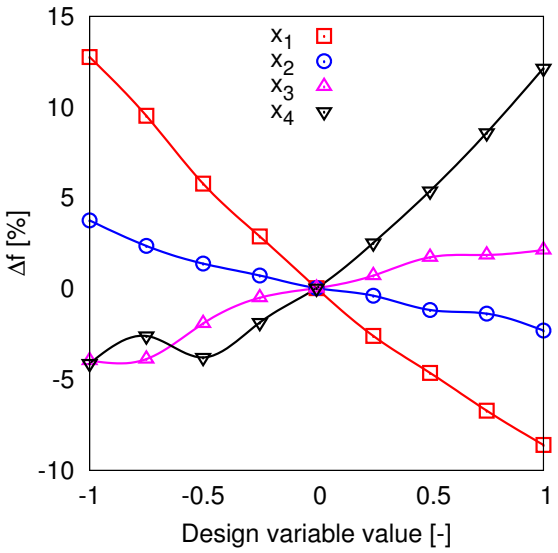


Figure 5.23: Problem I, sensitivity analysis

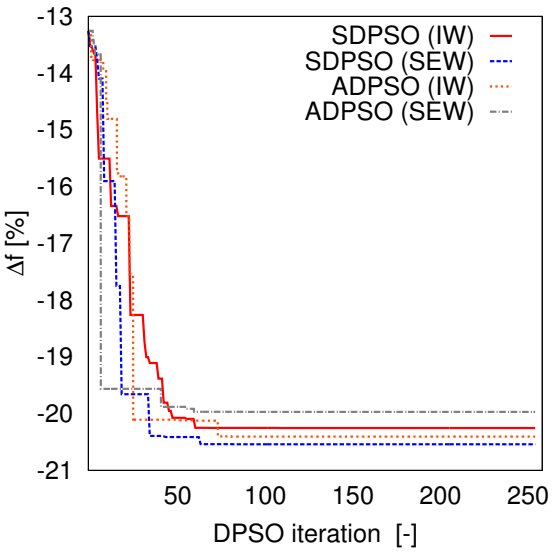


Figure 5.24: Problem I, convergence of S-DPSO and A-DPSO

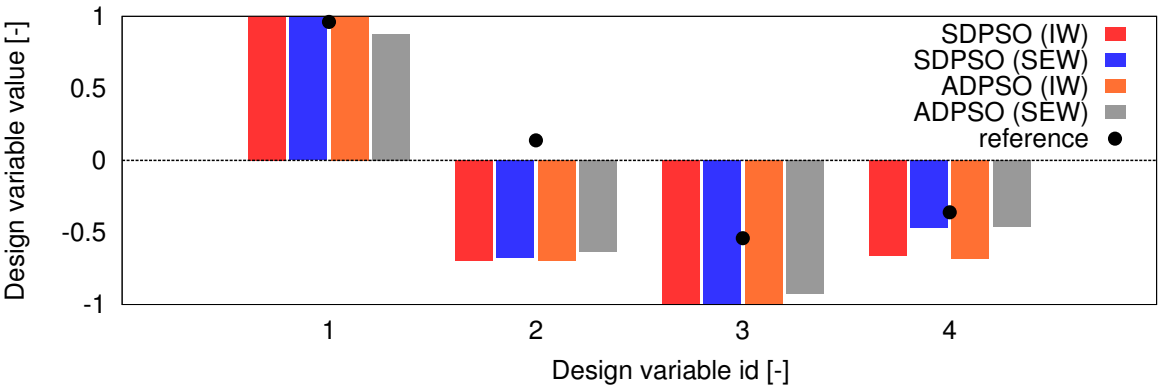


Figure 5.25: Problem I, comparison between optimal design variables of S-DPSO, A-DPSO with IW and SEW by PF and those obtained by metamodels with URANS [47].

Table 5.10: Problem I, SBD results

	Wall-type	R_T[N]	W[N]	f[−]	$\Delta f\%$
Original		50.15	852.5	5.88e-2	–
S-DPSO	IW	39.92	850.9	4.69e-2	-20.24
	SEW	40.36	850.6	4.67e-2	-20.57
A-DPSO	IW	39.85	851.1	4.68e-2	-20.41
	SEW	40.68	851.7	4.71e-2	-19.90

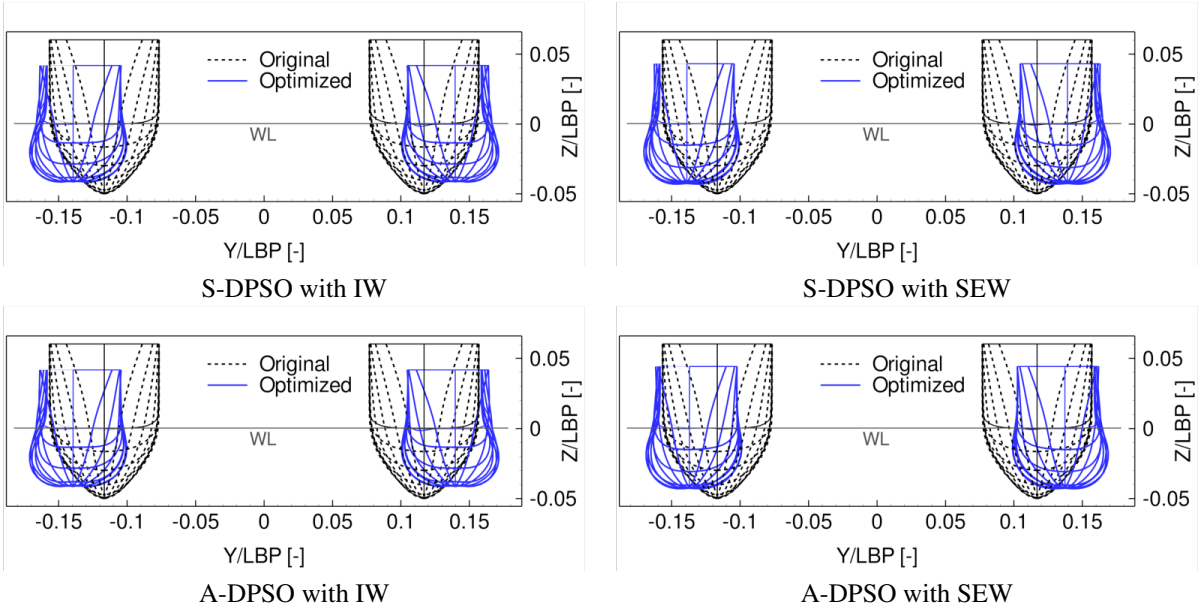


Figure 5.26: Problem I, comparison between optimized and original hull form

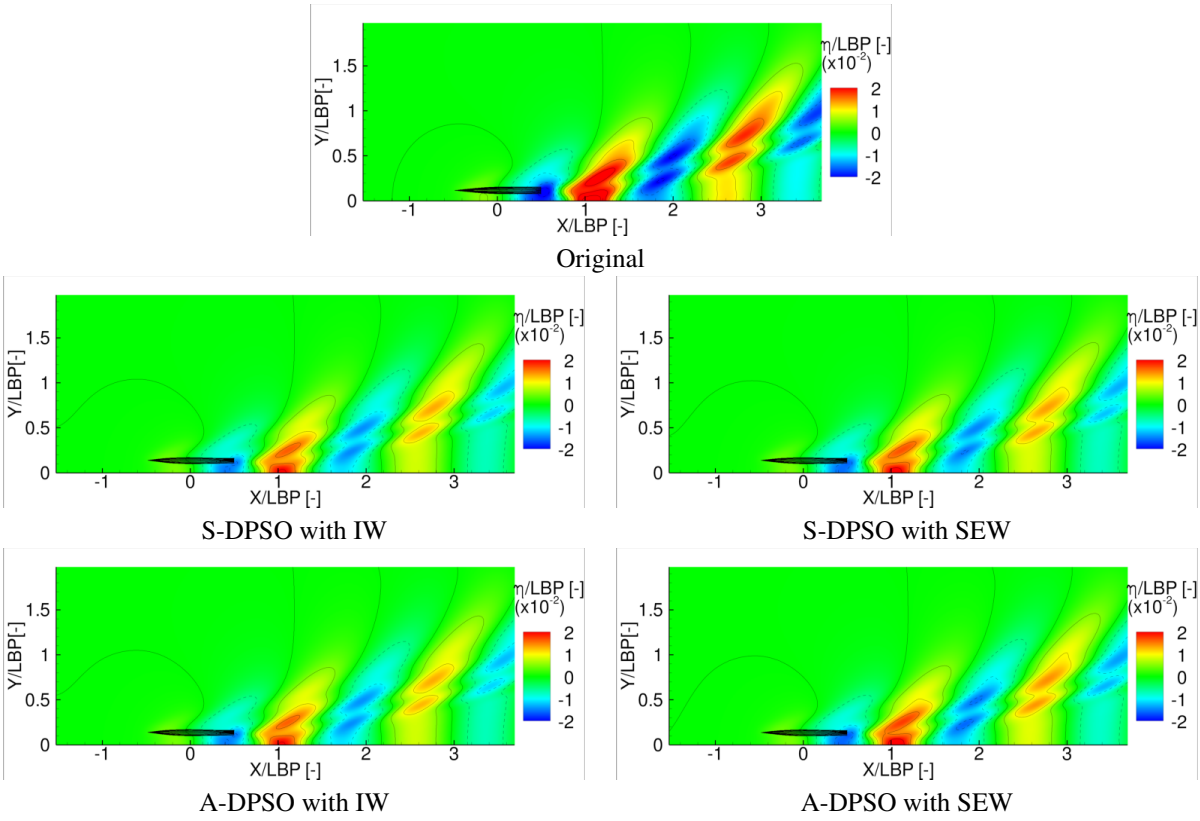


Figure 5.27: Problem I, comparison between optimized and original wave elevation pattern at Fr=0.5

Problem II

A preliminary sensitivity analysis for each design variable is shown in Fig. 5.28, showing $\Delta f(\%)$ compared to the parent hull. Changes in f are found significant in each direction, revealing a reduction of the objective function close to 10%.

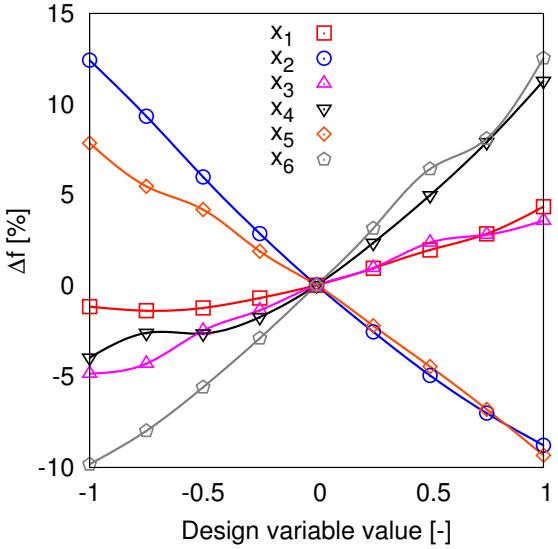


Figure 5.28: Problem II, sensitivity analysis

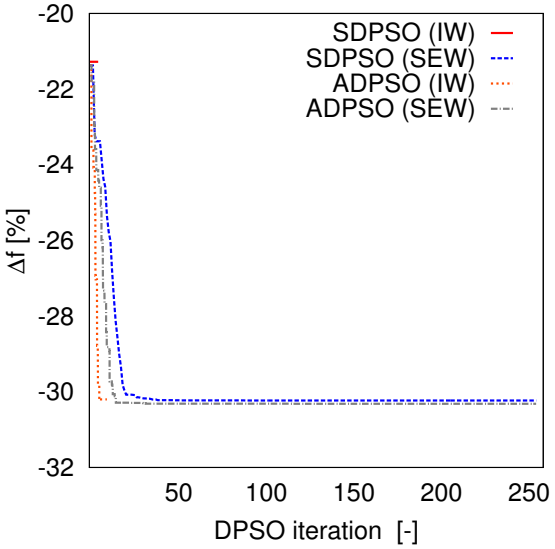


Figure 5.29: Problem II, convergence of S-DPSO and A-DPSO

The optimization is performed with both S-DPSO and A-DPSO, as per the guideline suggested in Tab. 5.5. Optimization results in Tab. 5.11 show that S-DPSO and A-DPSO with both IW and SEW lead to a reduction of the objective function greater than 20%. S-DPSO and A-DPSO iterations are shown in Fig. 5.29, revealing a quite sudden convergence. There are not significant differences between the results obtained by S-DPSO and A-DPSO except for S-DPSO with IW (see Figs. 5.30 and 5.31). In this case the IW approach induces the optimization to stop after 6 iterations. As shown in Fig. 5.30, differences in optimal design variables are mainly due to IW or SEW. Moreover, the optimum configuration leads to a considerable reduction of wave's elevation compared to the original shape (Figs. 5.32).

Finally, it may be noted that the final configuration is significantly different to that obtained using metamodels and a URANS solver [47], used here for reference (Fig. 5.30).

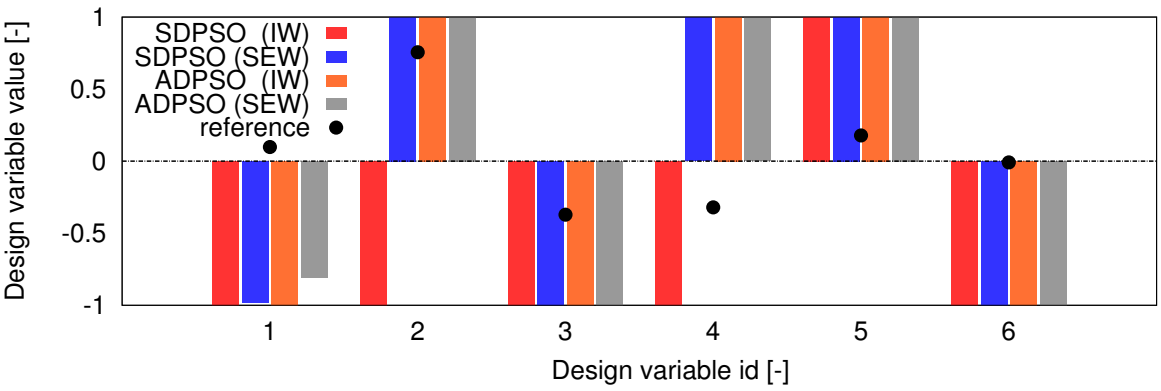


Figure 5.30: Problem II, comparison between optimal design variables of S-DPSO, A-DPSO with IW and SEW by PF and those obtained by metamodels with URANS [47].

Table 5.11: Problem II, SBD results

	Wall-type	R_T[N]	W[N]	f[−]	$\Delta f\%$
Original		50.15	852.5	5.88e-2	–
S-DPSO	IW	39.41	849.1	4.64e-2	-21.10
	SEW	34.29	835.4	4.10e-2	-30.27
A-DPSO	IW	34.31	835.5	4.11e-2	-30.10
	SEW	34.08	830.8	4.09e-2	-30.30

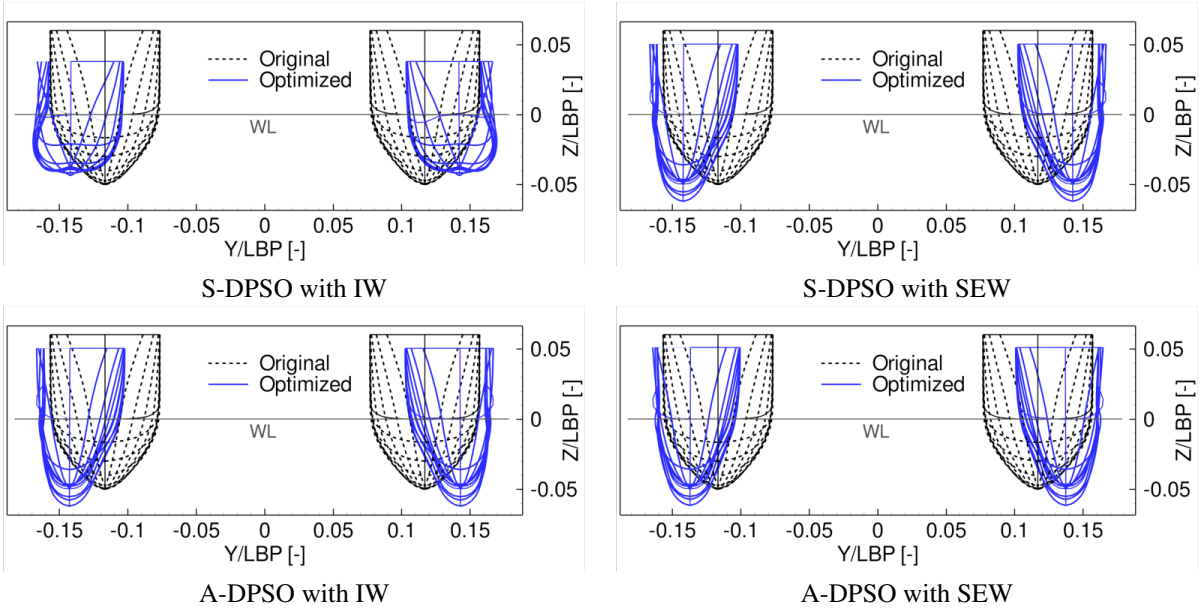


Figure 5.31: Problem II, comparison between optimized and original hull form

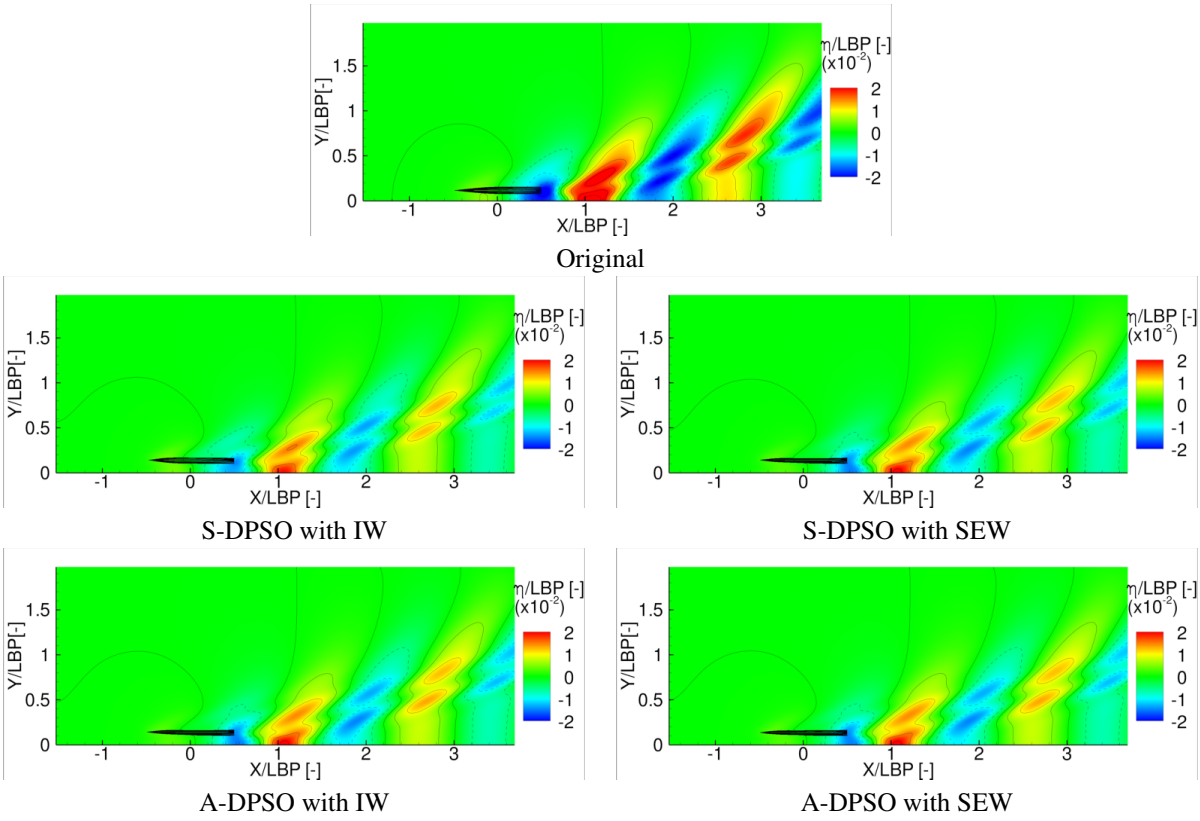
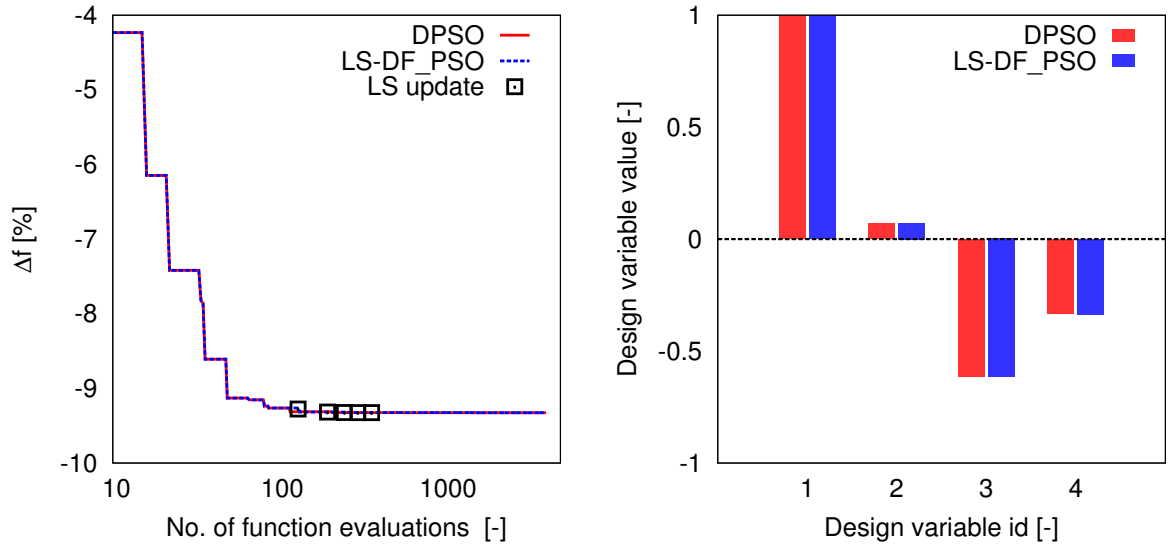


Figure 5.32: Problem II, comparison between optimized and original wave elevation pattern at Fr=0.5

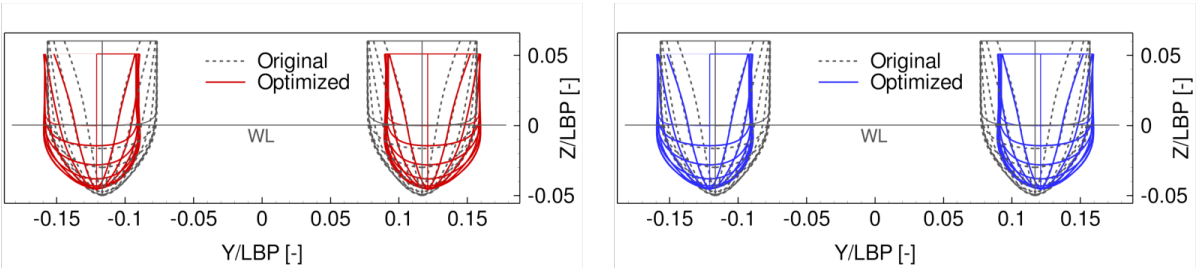
Problem III

Figure 5.33a shows the convergence of the minimization procedure, comparing DPSO with LS-DF_PSO. The line-search (LS) is required only 5 times, as indicated by the black squares, and the minima provided by the two algorithms are extremely close, as shown in Figs 5.33b and 5.34. Nevertheless, it may be noted that, at very reduced additional cost, LS-DF_PSO provides a solution certified with stationarity properties. Table 5.12 summarized the optimization results.



(a) (b)

Figure 5.33: Problem III, minimization of calm-water resistance for the Delft catamaran: (a) convergence of the objective function and (b) global-optimum design variables



(a) DPSO (b) LS-DF_PSO

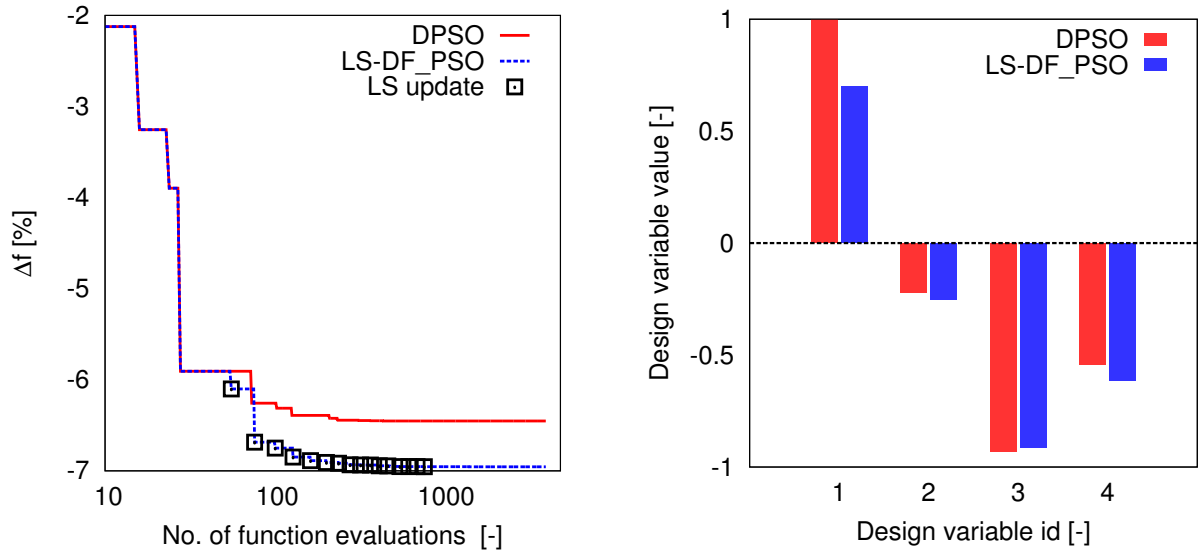
Figure 5.34: Problem III, comparison between optimized and original hull form

Table 5.12: Problem III, summary of the optimization results

	Design variable value (non-dimensional)				
Algorithm	x_1	x_2	x_3	x_4	$\Delta f\%$
DPSO	1.000	0.535	0.193	0.333	-9.326
LS-DF_PSO	1.000	0.536	0.192	0.332	-9.326

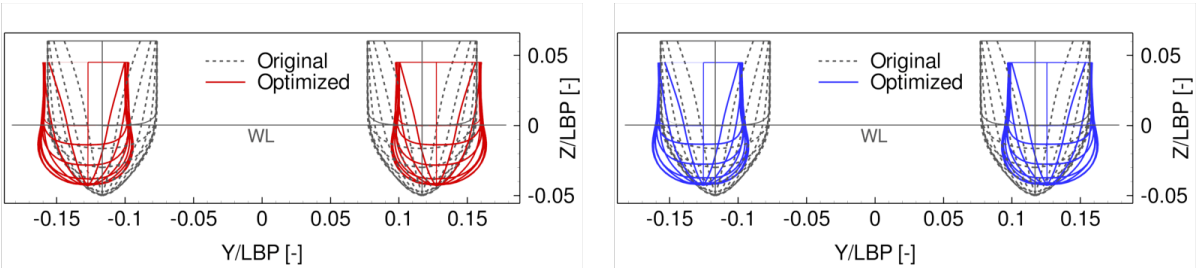
Problem IV

Figure 5.35a shows the convergence of the minimization procedure, comparing DPSO with LS-DF_PSO. LS is required and applied 16 times and is essential to identify the global optimum, as shown in Fig. 5.35b. As a results, optimal shape design provided by DPSO and LS-DF_PSO are noticeably different (within the context of current application's variation), as shown in Fig. 5.36. Additionally, it may be noted that the solution given by LS-DF_PSO is also endowed with stationarity properties. Table 5.13 summarizes the optimization results.



(a) (b)

Figure 5.35: Problem IV, minimization of expected value of mean resistance in head waves for the Delft catamaran: (a) convergence of the objective function and (b) global-optimum design variables



(a) DPSO (b) LS-DF_PSO

Figure 5.36: Problem IV, comparison between optimized and original hull form

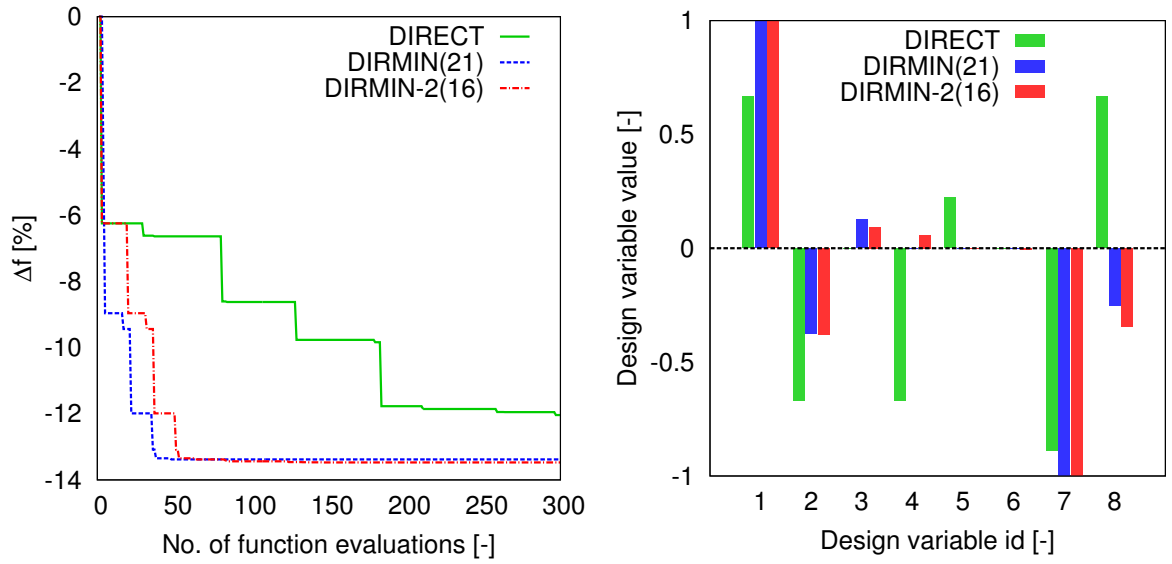
Table 5.13: Problem IV, summary of the optimization results

Algorithm	Design variable value (non-dimensional) x_1	x_2	x_3	x_4	$\Delta f\%$
DPSO	1.000	0.389	0.034	0.228	-6.453
LS-DF_PSO	0.849	0.374	0.043	0.192	-6.956

5.2.2 DTMB 5415

Problem V

The SBDO procedure achieves a reduction of the objective function value by 12.04%, 13.38%, and 13.47% using DIRECT, DIRMIN, and DIRMIN-2, respectively, as suggested in Sec. 5.1.3. The convergence history of the objective function is shown in Fig. 5.37a, confirming the efficiency of the two global/local hybrid methods, for a very limited budget of function evaluations ($N_{\max} = 300$). DIRMIN and DIRMIN-2 achieve almost the final objective function reduction in the first 50 evaluations, whereas DIRECT is not able to reach the same value within the imposed limit of 300 function evaluations.

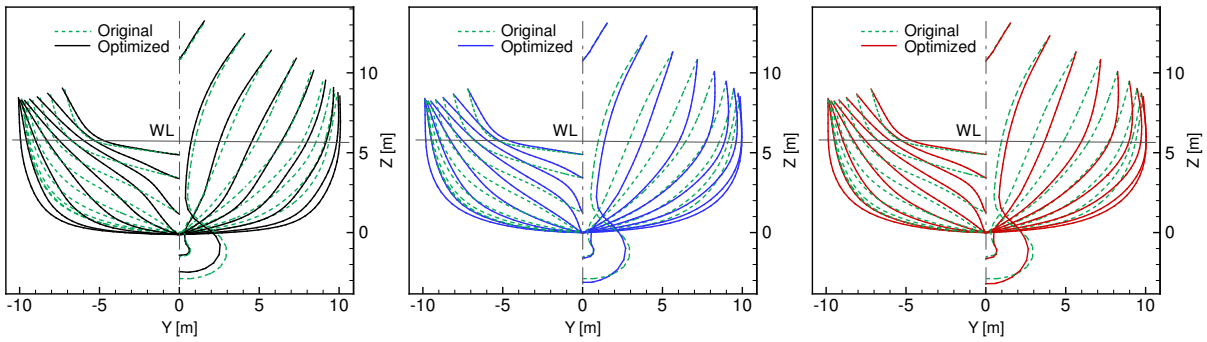


(a) (b)

Figure 5.37: Problem V, (a) objective function convergence history and (b) optimal design variables

Figure 5.37b presents the values of the corresponding optimal design variables, showing appreciable differences for DIRECT, DIRMIN, and DIRMIN-2. Optimal design variable value and objective function reductions are summarized in Tab. 5.14. Figure 5.38 shows the sections of the optimized hull compared to the original. The reduction of the total resistance is consistent with the reduction of the wave elevation patterns, both in terms of transverse and diverging Kelvin waves, as shown in Fig.5.39. Figure 5.39 shows also the pressure field on the optimized hulls compared to the original, showing a slightly better pressure recovery towards the stern.

Finally, Tab. 5.15 summarizes the main parameters associated with the optimal DIRECT, DIRMIN, and DIRMIN-2 designs. The resistance coefficients are defined as $C_x = R_x/0.5\rho U^2 S_{w,stat}$, with R_w, R_f, R_T being the wave, frictional and total resistance, respectively; $S_{w,stat}$ and $S_{w,dyn}$ are the static and dynamic wetted surface areas.



(a) DIRECT (b) DIRMIN (a) DIRMIN-2

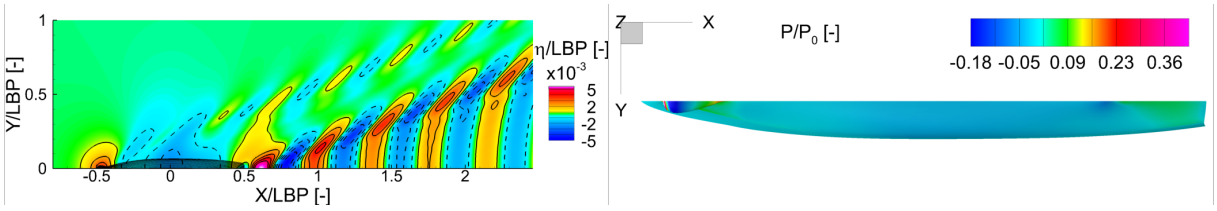
Figure 5.38: Problem V, optimal hull-form shapes compared with the original for Fr=0.25

Table 5.14: Problem V, summary of optimization results for DTMB 5415 model

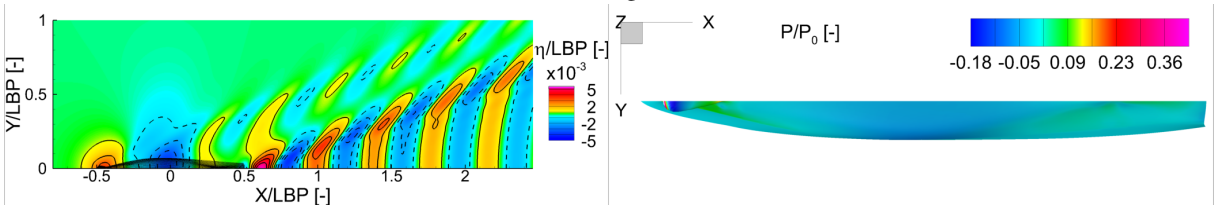
$N_{\max}$	Algorithm	Design variables values (non-dimensional) x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$R_T(\times 10^5)$[N] value	$\Delta f\%$
	DIRECT	0.667	-0.667	0.000	-0.667	0.222	0.000	-0.889	0.667	3.021	-12.04
300	DIRMIN	1.000	-0.375	0.125	0.000	0.000	0.000	-1.000	-0.250	2.974	-13.38
	DIRMIN-2	1.000	-0.379	0.094	0.059	0.000	-0.008	-1.000	-0.344	2.972	-13.47

Table 5.15: Problem V, summary of optimization results for DTMB 5415 hull form

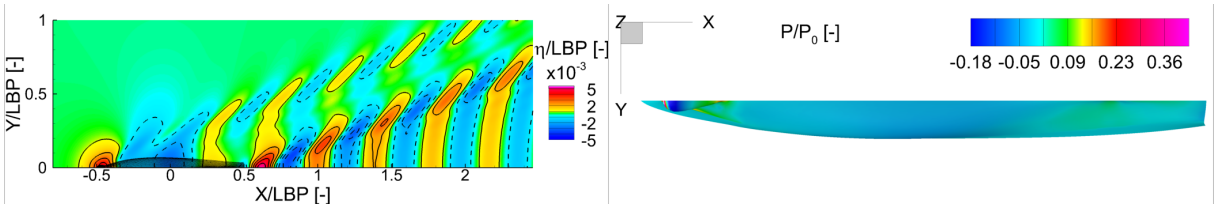
Parameter	Original value	unit	Optimized $\Delta\%_{\text{orig}}$ DIRECT	DIRMIN	DIRMIN-2
C_w	1.04e-3	–	-28.3	-33.2	-33.9
C_f	1.60e-3	–	0.05	-0.06	-0.13
C_T	2.64e-3	–	-11.3	-13.2	-13.5
σ/LBP	9.42e-4	–	6.82	10.1	9.73
τ	8.99e-4	rad	105.4	150.2	144.7
$S_{w,\text{stat}}/LBP^2$	1.48e-1	–	-0.85	-0.19	0.03
$S_{w,\text{dyn}}/LBP^2$	1.50e-1	–	-0.80	-0.25	-0.02



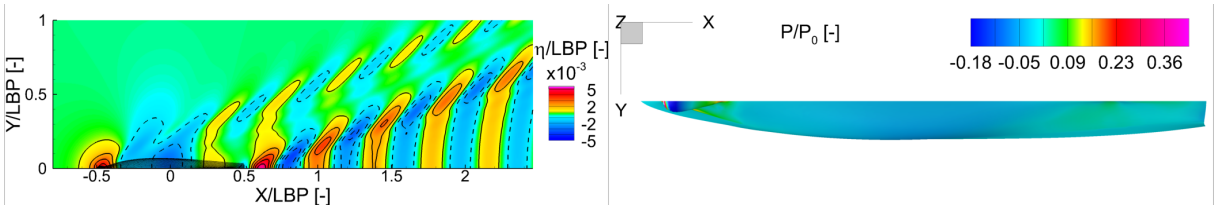
(a) Original



(b) DIRECT



(c) DIRMIN



(d) DIRMIN-2

Figure 5.39: Problem V, wave elevation pattern (left) and pressure field distribution (right) of the optimized hulls, compared to the original (a) for Fr=0.25

Problem VI

A preliminary sensitivity analysis for each design variable is presented in Fig. 5.40, showing the associated percent resistance reduction ($\Delta f\%$) with respect to the original hull. Non-dimensional design variables are shown in the plots. Changes in f are found significant for all variables but x_5 and x_6, revealing a possible reduction of the total resistance at Fr = 0.25 close to 10%. The analysis of the results is conducted setting apart results (i) for a low budget of 192 function evaluations (which corresponds to $32N$, an eighth of the full budget), and (ii) for the full budget of 1536 function evaluations (which corresponds to $256N$).

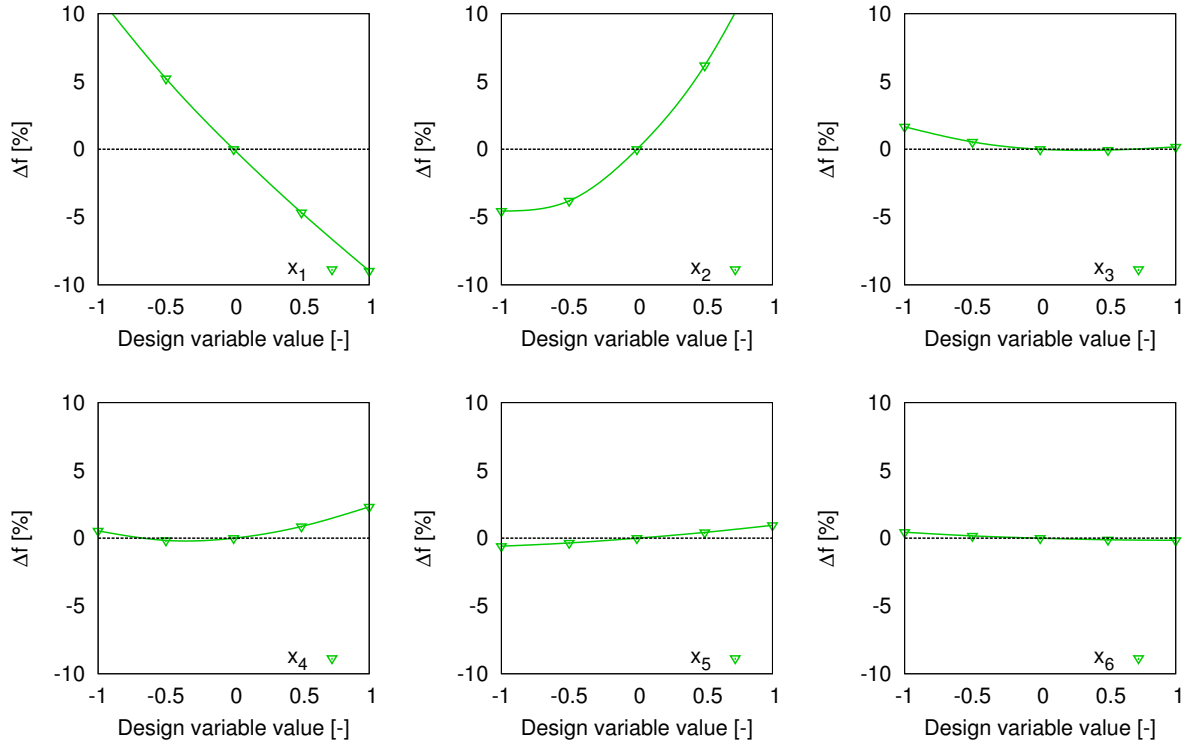
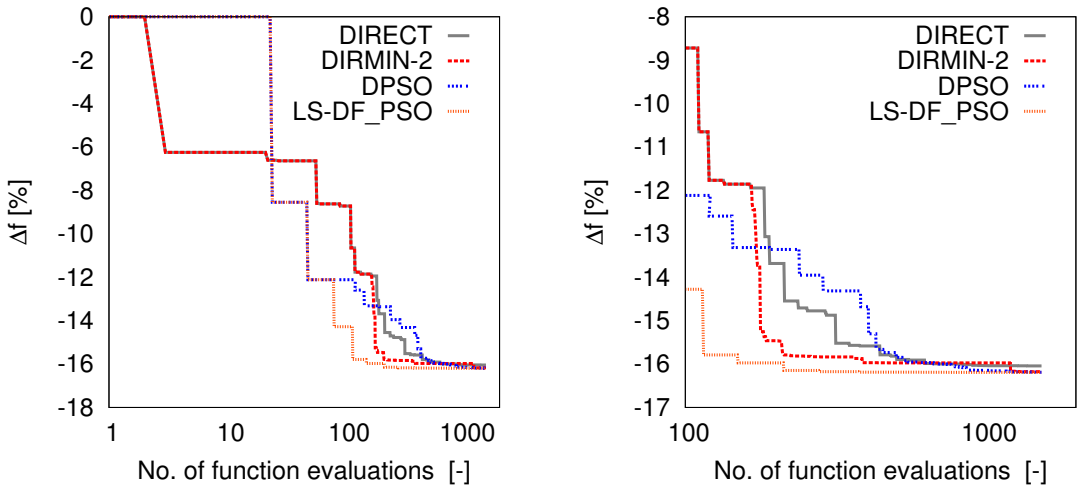


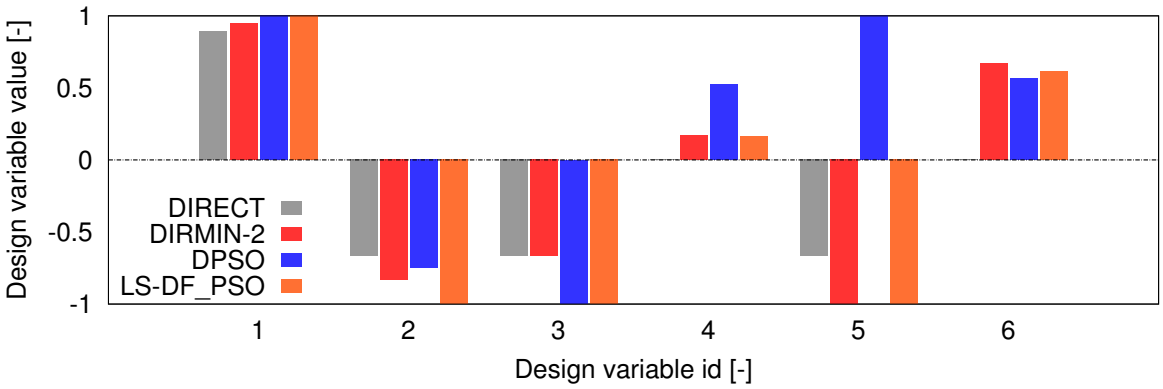
Figure 5.40: Problem VI, sensitivity analysis of non-dimensional design variables



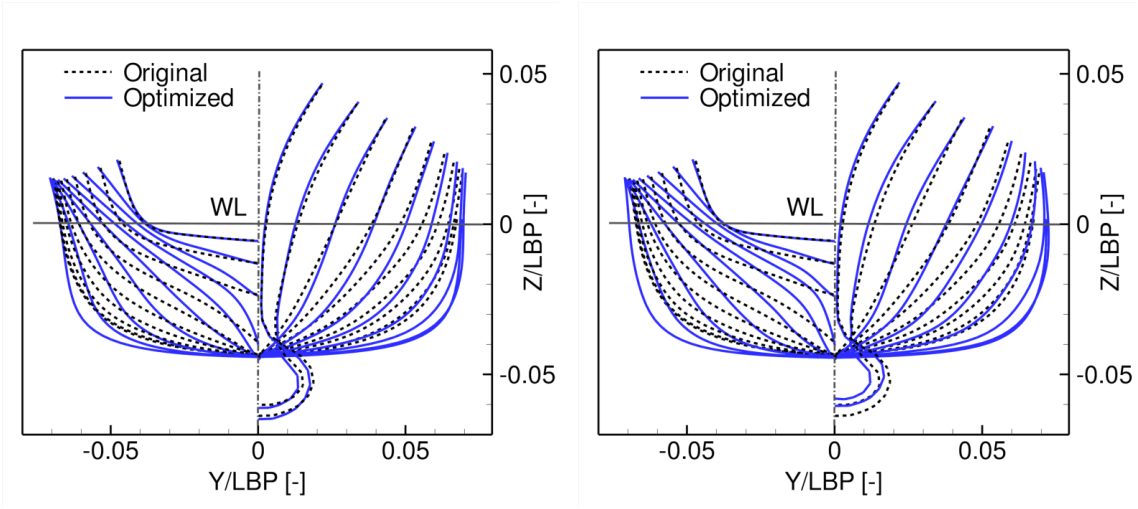
(a) (b)

Figure 5.41: Problem VI, (a) objective function convergence history and (b) detail after the first 100 function evaluations

For the case (i), the optimization procedure achieves a resistance reduction of 13.7% and 15.5% using DIRECT and DIRMIN-2 respectively, and a reduction of 13.5% and 16.0% using DPSO and LS-DF_PSO respectively. The two global/local hybrid algorithms outperform their global version. In particular, LS-DF_PSO is found the most efficient algorithm for the present SBD problem, achieving the best design with the fastest convergence rate, as shown in Fig. 5.41. Figure 5.42 presents the values of the optimized design variables, showing appreciable differences, and illustrates the corresponding optimized shapes, compared to the original.

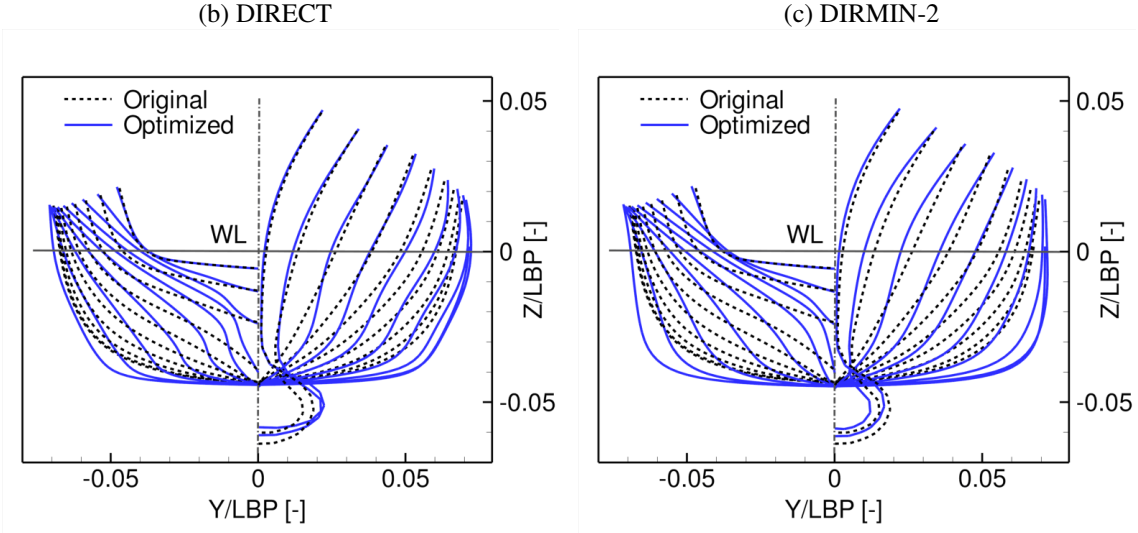


(a) Objective function convergence of optimized design variables



(b) DIRECT

(c) DIRMIN-2

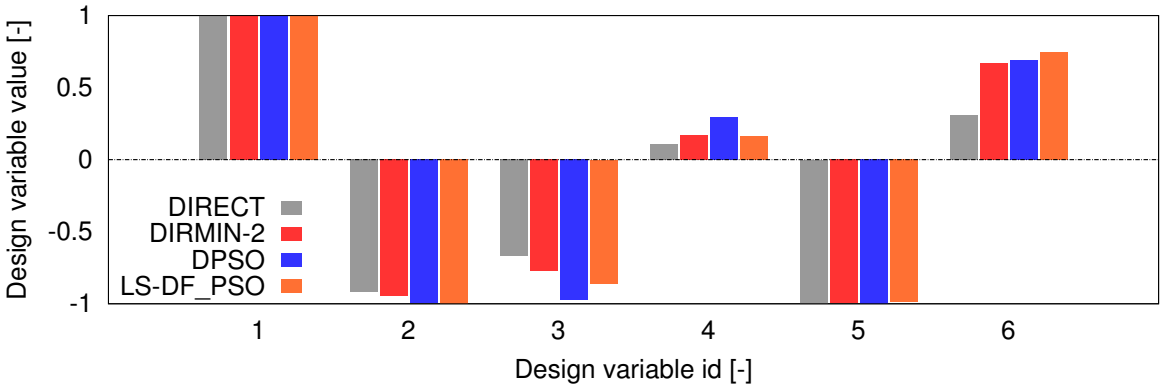


(d) DPSO

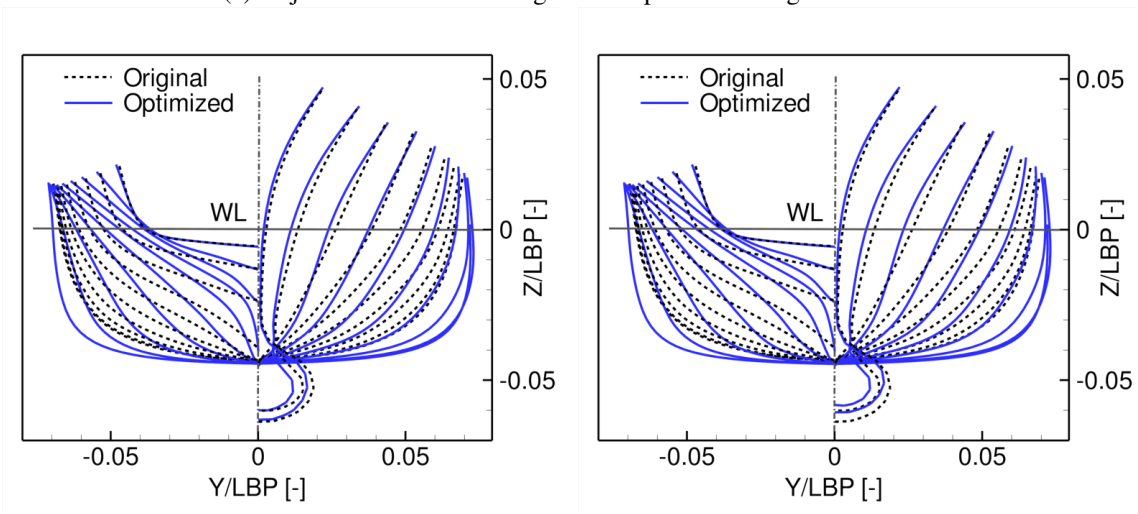
(e) LS-DF_PSO

Figure 5.42: Problem VI, 192 function evaluations

For the case (ii), the optimization procedure achieves a resistance reduction of 16.0% and 16.2% using DIRECT and DIRMIN-2 respectively, and a reduction of 16.2% using both DPSO and LS-DF_PSO. The convergence history of the objective function towards the minimum is shown in Fig. 5.41, confirming the efficiency and robustness of the two hybrid global/local approaches DIRMIN-2 and LS-DF_PSO. More in detail, LS-DF_PSO achieves the most significant reduction of the objective function overall, although all the solutions are very close in this case. Figure 5.43 presents the values of the corresponding optimized design variables and shows the optimized shapes compared to the original. The close agreement of the solutions obtained by the different algorithms indicates that the global minimum region has been likely achieved. A summary of the optimization results is presented in Tab. 5.16.

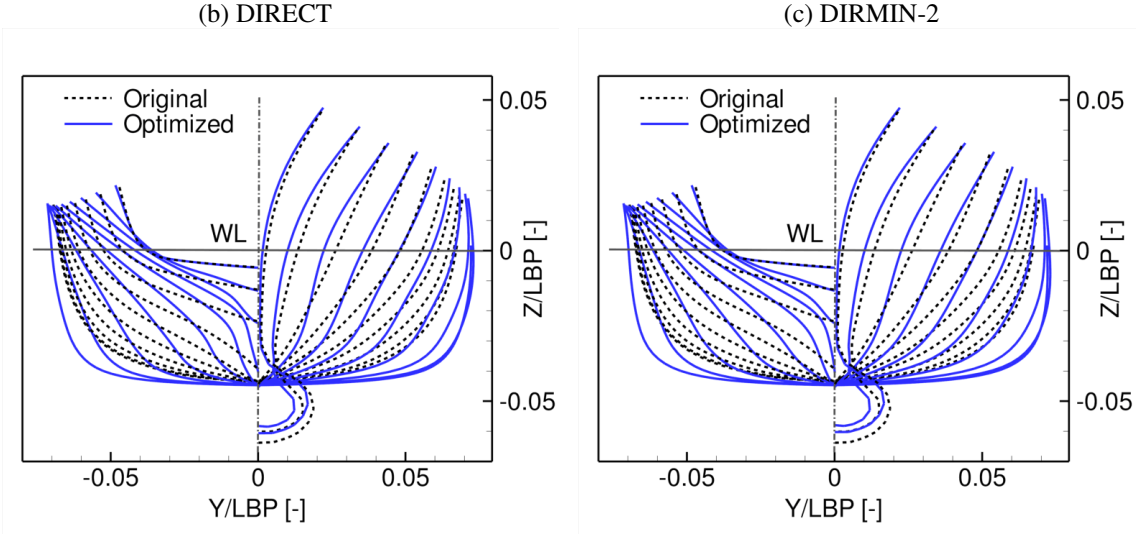


(a) Objective function convergence of optimized design variables



(b) DIRECT

(c) DIRMIN-2



(d) DPSO

(e) LS-DF_PSO

Figure 5.43: Problem VI, 1536 function evaluations

The reduction of the wave elevation pattern of the final shape, both in terms of transverse and diverging stern waves, is visible in Fig. 5.44. Finally, Fig. 5.45 presents the pressure field on the optimized hulls compared to the original hull, showing a better pressure recovery towards the stern. Obviously, the solver is not able to predict flow separations, which are likely to occur for such large design modifications.

Table 5.16: Problem VI, summary of the optimization results

N_{max}	Algorithm	Design variables values (non-dimensional) x_1	x_2	x_3	x_4	x_5	x_6	$R_T(\times 10^5)$ [N] value	$\Delta f\%$
192	DIRECT	0.89	-0.67	-0.67	0.00	-0.67	0.00	2.96	-13.7
	DIRMIN-2	0.94	-0.83	-0.67	0.17	-1.00	0.67	2.92	-15.5
	DPSO	1.00	-0.75	-1.00	0.52	1.00	0.56	2.96	-13.5
	LS-DF_PSO	1.00	-1.00	-1.00	0.16	-1.00	0.61	2.89	-16.0
1536	DIRECT	1.00	-0.92	-0.67	0.11	-1.00	0.31	2.88	-16.0
	DIRMIN-2	1.00	-0.94	-0.77	0.17	-1.00	0.67	2.88	-16.2
	DPSO	1.00	-0.99	-0.97	0.29	-1.00	0.69	2.88	-16.2
	LS-DF_PSO	1.00	-1.00	-0.86	0.16	-0.99	0.75	2.88	-16.2

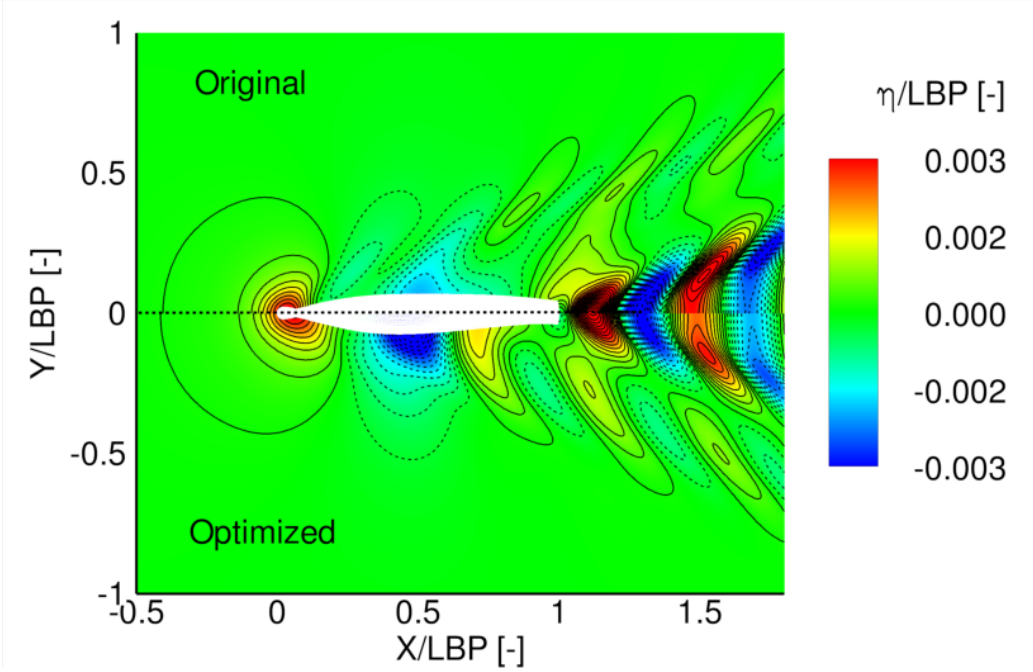


Figure 5.44: Problem VI, wave patterns produced by LS-DF_PSO optimized hull forms at Fr = 0.25 compared with original

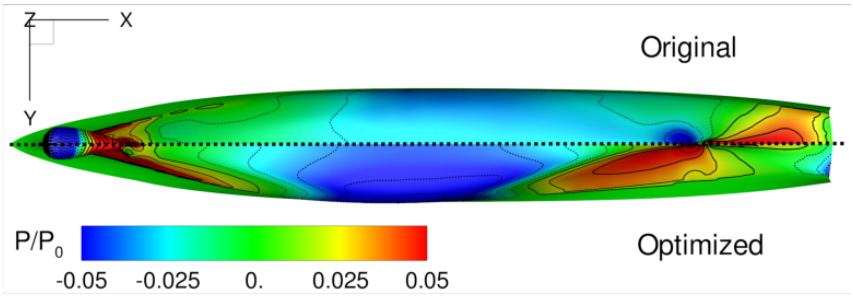


Figure 5.45: Problem VI, pressure field on LS-DF_PSO optimized hull forms at Fr = 0.25 compared with original

Problem VII

A preliminary sensitivity analysis for each design variable is presented in Fig. 5.46, showing the associated percent resistance reduction (Δf) with respect to the original hull.

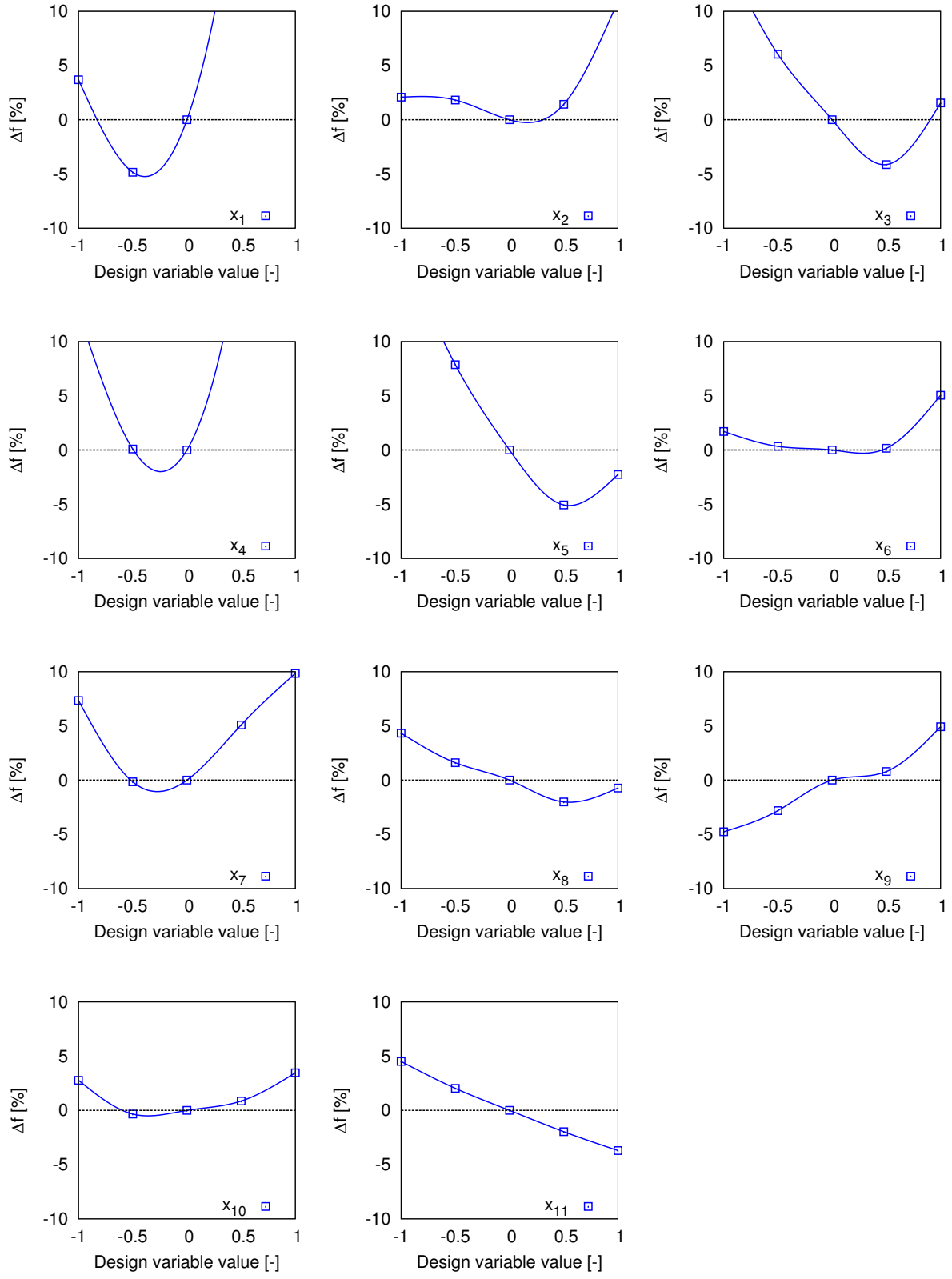
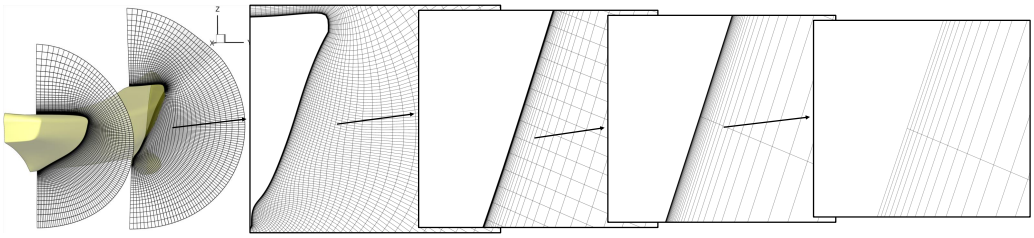
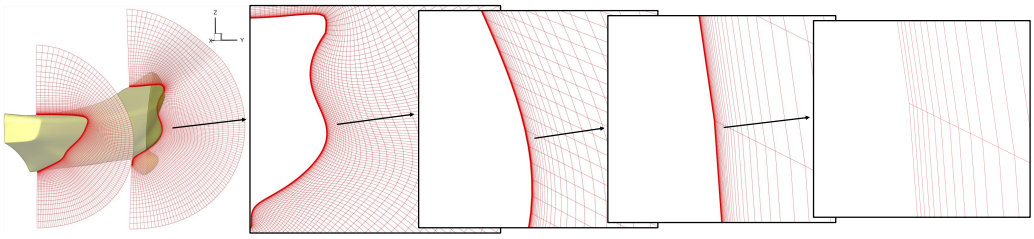


Figure 5.46: Problem VII, sensitivity analysis of non-dimensional design variables

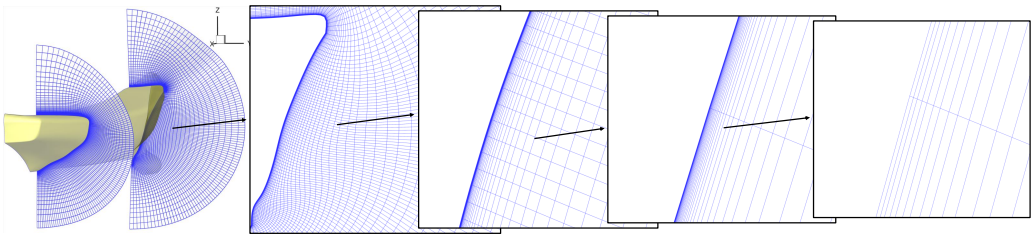
The quality of the grids produced by the method presented in Sec. 4.2.2 is assessed for each modified design. Figure 5.47b presents a modified grid laying on the boundary of the design space, showing an acceptable quality.



(a) Original hull



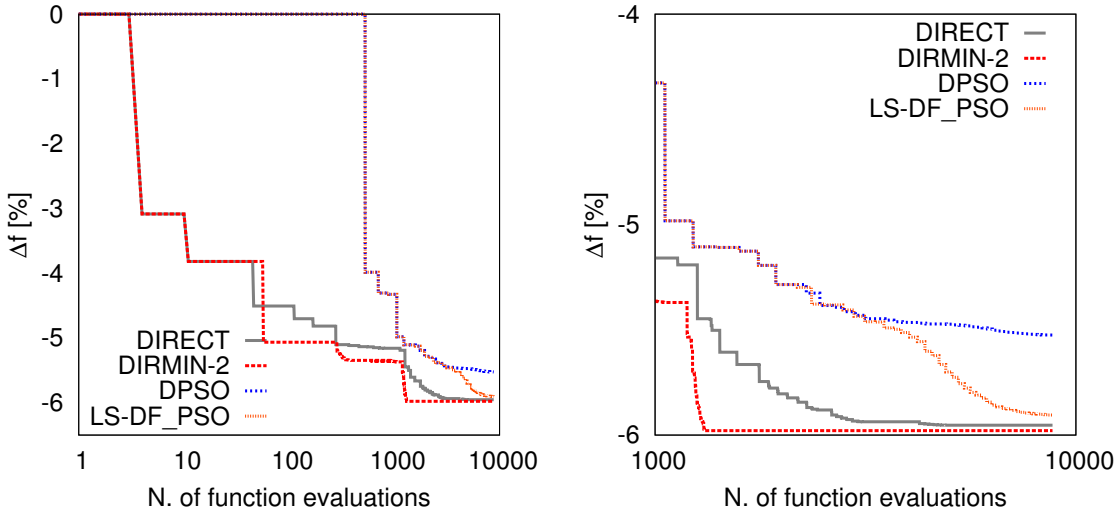
(b) Modified hull laying on the design space boundary ($x_1 = 1$)



(c) Optimized hull

Figure 5.47: Close view of the boundary layer grid (sections at constant I)

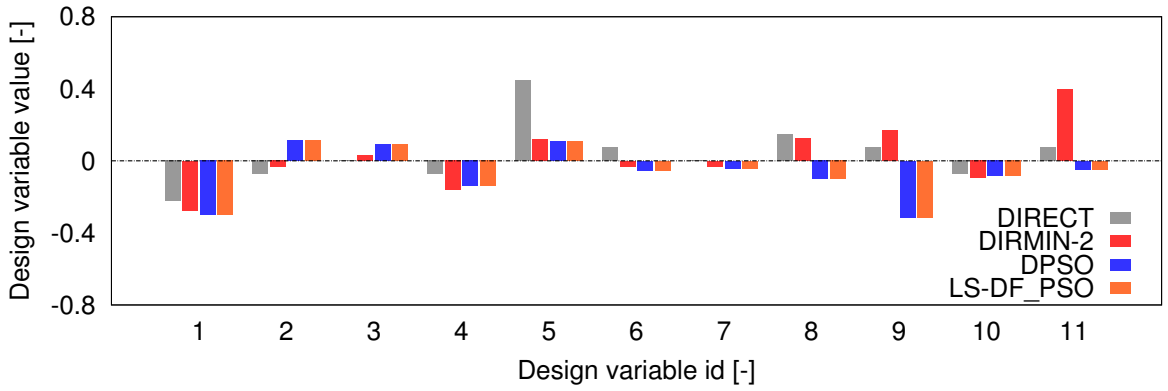
Non-dimensional design variables are shown in the plots. Changes in f are found significant for all variables, revealing a possible reduction of the total resistance at Fr = 0.25 close to 5%. Variations are smaller than Problem VI, due to more realistic simulations by the RANS solver. The analysis of the results is conducted setting apart results (i) for a low budget of 1100 function evaluations (which corresponds to $100N$, an eighth of the full budget), and (ii) for the full budget of 8800 function evaluations (which corresponds to $800N$).



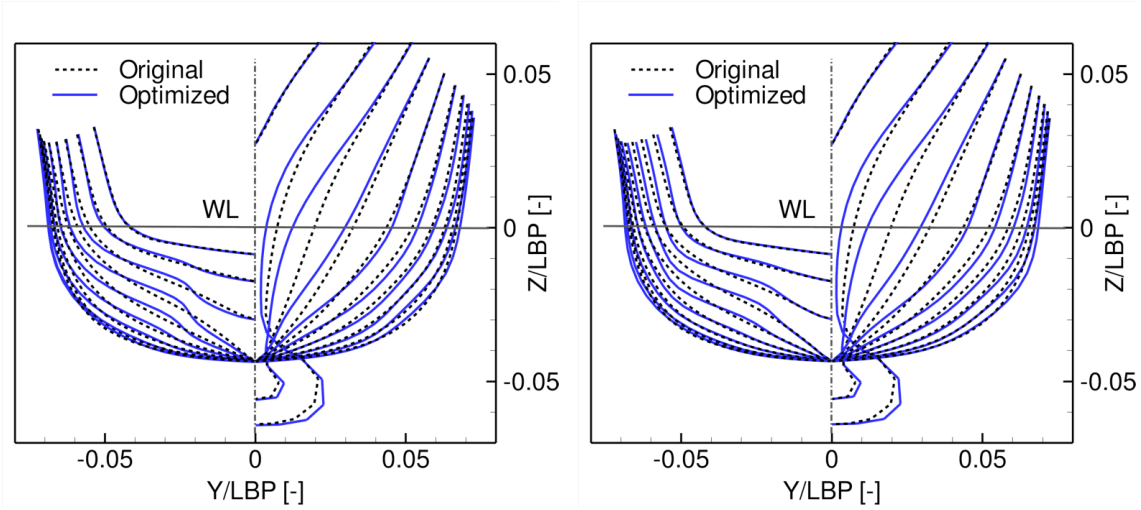
(a) (b)

Figure 5.48: Problem VII, (a) objective function convergence history and (b) detail after the first 1000 function evaluations

For the case (i), the optimization procedure achieves a total resistance coefficient reduction of 5.16% and 5.37% using DIRECT and DIRMIN-2 respectively, and a reduction of 4.98% using both DPSO and LS-DF_PSO. DIRMIN-2 outperform its global version, whereas DPSO and its hybrid reach the same result because non a local search has been activated by LS-DF_PSO. DIRMIN-2 is found the most efficient algorithm for the present SBD problem, achieving the best design with the fastest convergence rate, as shown in Fig. 5.48. Figure 5.49 presents the values of the optimized design variables, showing appreciable differences, and shows the corresponding optimized shapes, compared to the original.

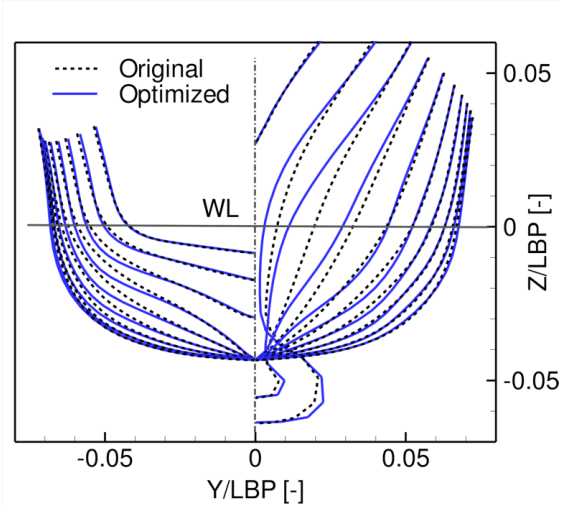


(a) Objective function convergence of optimized design variables

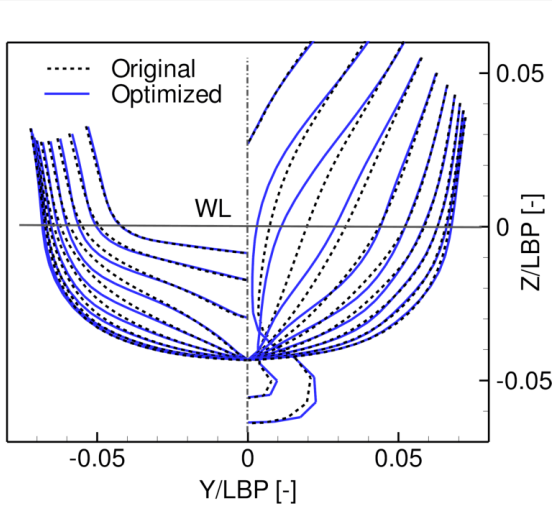


(b) DIRECT

(c) DIRMIN-2



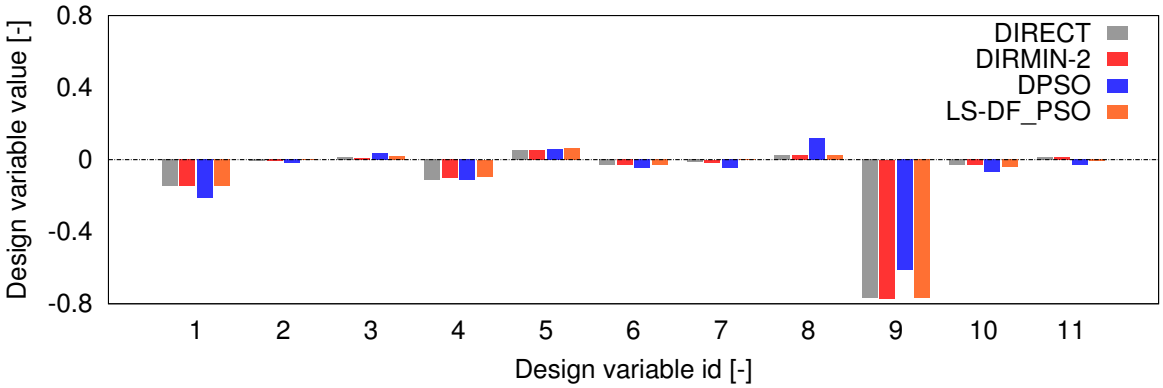
(d) DPSO



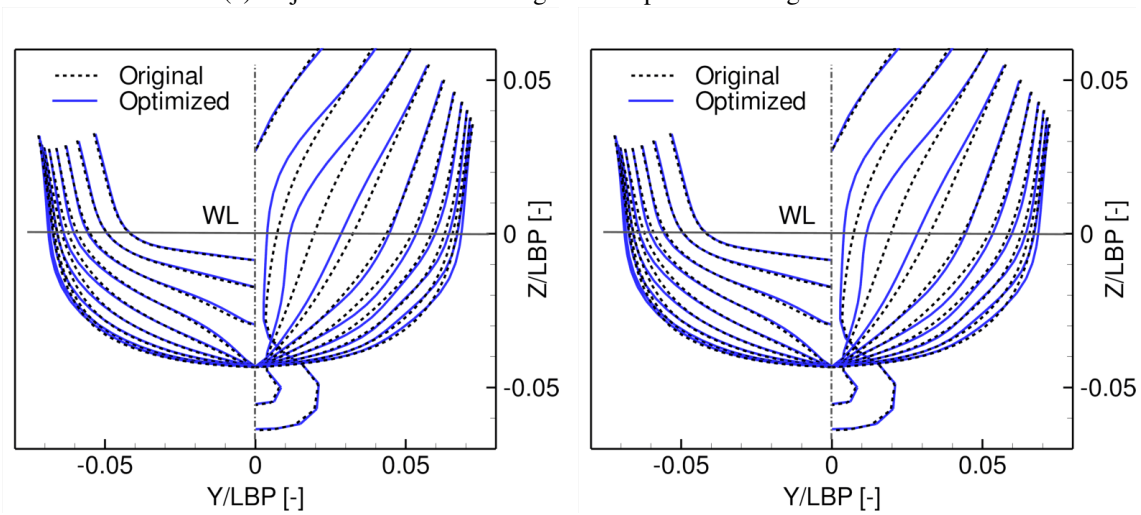
(e) LS-DF_PSO

Figure 5.49: Problem VII, 1100 function evaluations

For the case (ii), the optimization procedure achieves a resistance reduction of 5.95% and 5.98% using DIRECT and DIRMIN-2 respectively, and a reduction of 5.52% and 5.91% using DPSO and LS-DF_PSO respectively. The convergence history of the objective function towards the minimum is shown in Fig. 5.48, confirming the efficiency and robustness of the two hybrid global/local approaches DIRMIN-2 and LS-DF_PSO. More in detail, DIRMIN-2 achieves the most significant reduction of the objective function overall, although all the solutions are very close in this case. Figure 5.50 presents the values of the corresponding optimized design variables and shows the optimized shapes compared to the original. The close agreement of the solutions obtained by the different algorithms indicates that the global minimum region has been likely achieved. A summary of the optimization results is presented in Tab. 5.17.



(a) Objective function convergence of optimized design variables



(b) DIRECT

(c) DIRMIN-2

(d) DPSO

(e) LS-DF_PSO

Figure 5.50: Problem VII, 8800 function evaluations

The best design is finally assessed with RANS. The associated modified grid is assessed and presented in Fig. 5.47c, showing a good quality. The results are presented in Figs. 5.51 and 5.52, and Tab. 5.18. Figure 5.51 show a significant reduction of the diverging bow wave and a small reduction of the diverging and transverse stern wave. It may be also noted how the shoulder wave is cancelled. Specifically, the optimized shoulder shape induces a high pressure region in correspondence of the first trough of the original hull, causing a phase shift with the reduction of the diverging bow wave and the cancellation of the shoulder wave (well visible in Fig. 5.51). This effect has been also shown in retrofitting studies by optimization of blisters attached to the original hull [135]. Moreover, a more uniform pressure distribution along the hull and a slightly better pressure recovery at the stern for the optimized hull (see Fig. 5.51). A longitudinal wave cut along the $y = 0.1LBP$ plane is shown in Fig. 5.52, highlighting the reduction of the wave elevation, especially at the bow.

The hydrodynamic coefficients for the original and the optimized hulls are finally compared in Tab. 5.18, confirming that a large part of the resistance reduction stems from the reduction of the piezometric pressure coefficient.

Table 5.17: Problem VII, summary of the optimization results

		Design variables values (non-dimensional)											$C_T\,(\times 10^{-3})$ [–]	
$N_{\max}$	Algorithm	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	value	$\Delta f\%$
1100	DIRECT	-0.22	-0.07	0.00	-0.07	0.44	0.07	0.00	0.15	0.07	-0.07	0.07	4.00	-5.16
	DIRMIN-2	-0.28	-0.03	0.03	-0.16	0.12	-0.03	-0.03	0.13	0.17	-0.09	0.40	3.99	-5.37
	DPSO	-0.30	0.11	0.09	-0.14	0.11	-0.05	-0.04	-0.10	-0.31	-0.08	-0.05	4.01	-4.98
	LS-DF_PSO	-0.30	0.11	0.09	-0.14	0.11	-0.05	-0.04	-0.10	-0.31	-0.08	-0.05	4.01	-4.98
8800	DIRECT	-0.15	-0.01	0.01	-0.11	0.05	-0.03	-0.01	0.02	-0.77	-0.03	0.01	3.97	-5.95
	DIRMIN-2	-0.15	-0.01	0.01	-0.10	0.05	-0.03	-0.02	0.02	-0.77	-0.03	0.02	3.97	-5.98
	DPSO	-0.21	-0.02	0.04	-0.11	0.06	-0.05	-0.04	0.12	-0.61	-0.07	-0.03	3.99	-5.52
	LS-DF_PSO	-0.15	-0.06	0.02	-0.09	0.06	-0.03	-0.03	0.03	-0.77	-0.04	-0.01	3.97	-5.91

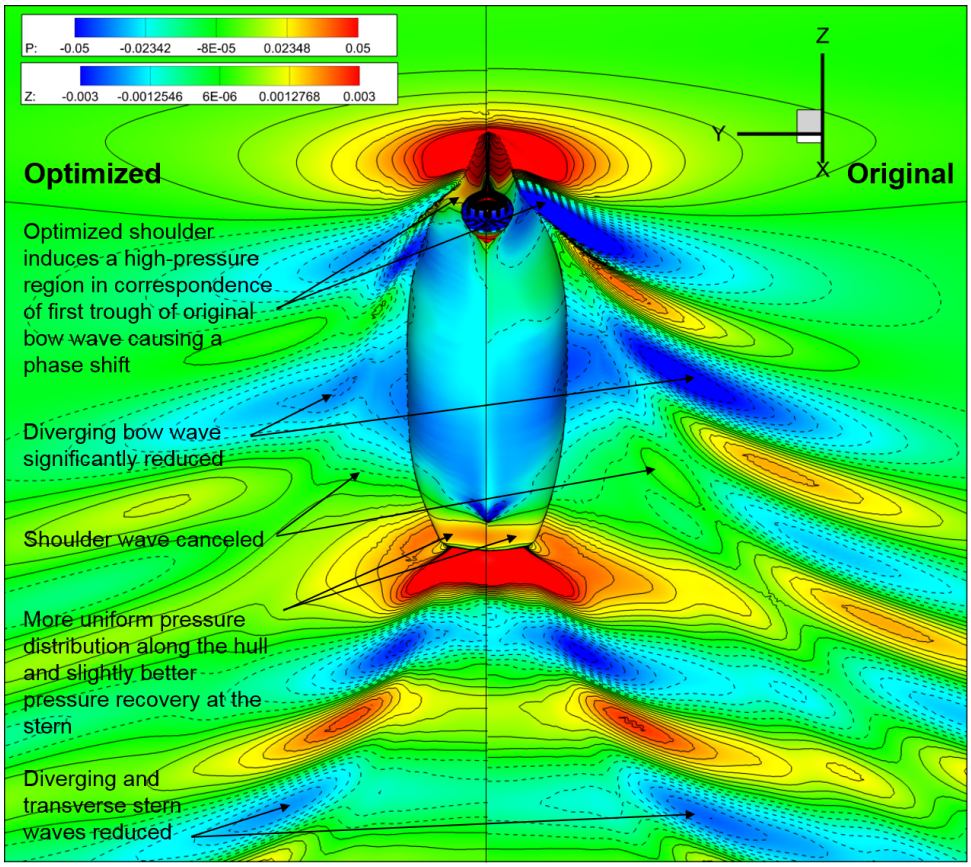


Figure 5.51: Problem VII, bottom view of wave pattern and pressure distribution at Fr = 0.25 for optimized (left) and original (right) hulls

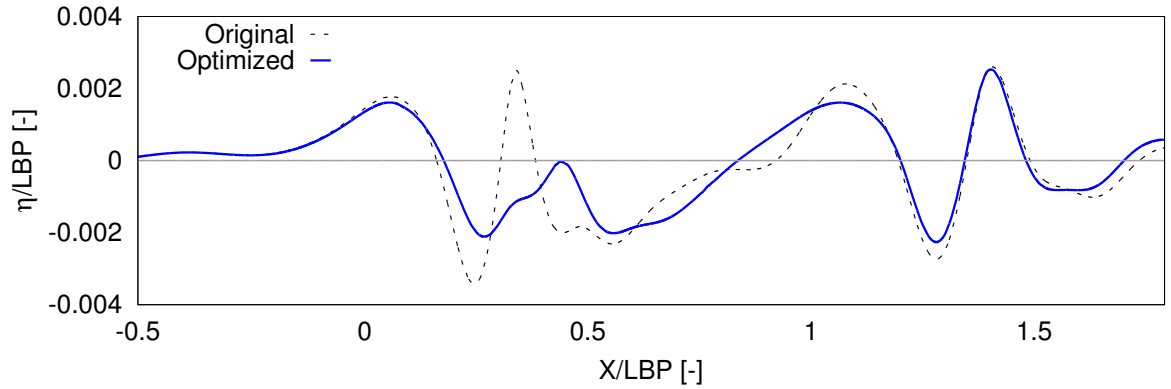


Figure 5.52: Problem VII, longitudinal wave cut on the $y = 0.1LBP$ plane at Fr = 0.25 for original and optimized hulls

Table 5.18: Problem VII, comparison between original and optimized DTMB 5415 hydrodynamic coefficients (C_{pp} represent the piezometric pressure, C_h the hydrostatic pressure, C_f the viscous shear stress, $C_{mg,x}$ the component of the weight force along the longitudinal axis, and C_T the total resistance)

Parameter	Unit	Original	Optimized	Δ%
C_{pp}	–	1.38e-3	9.08e-4	-34.0
C_h	–	0.86e-3	1.24e-3	42.0
C_f	–	3.16e-3	3.18e-3	0.65
$C_{mg,x}$	–	-1.19e-3	-1.35e-3	-13.4
C_T	–	4.21e-3	3.97e-3	-6.00
σ/LBP	–	-1.31e-3	-1.35e-3	-3.29
τ	deg	-0.11	-0.12	-15.3
$S_{w,stat}/LBP^2$	–	1.48e-2	1.50e-2	0.96

Problem VIII

Figure 5.53 shows a preliminary sensitivity analysis, varying the design variables vector along each KL modes. The direction associated with the first KL mode reveals the largest objective function reduction, close to 10%.

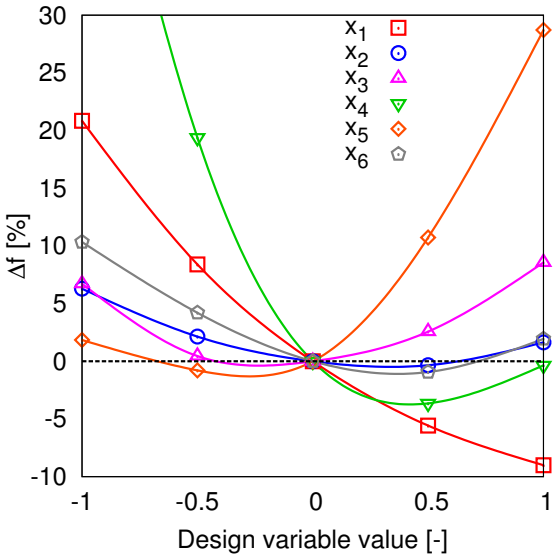


Figure 5.53: Problem VIII, sensitivity analysis for the objective function, along the KL modes

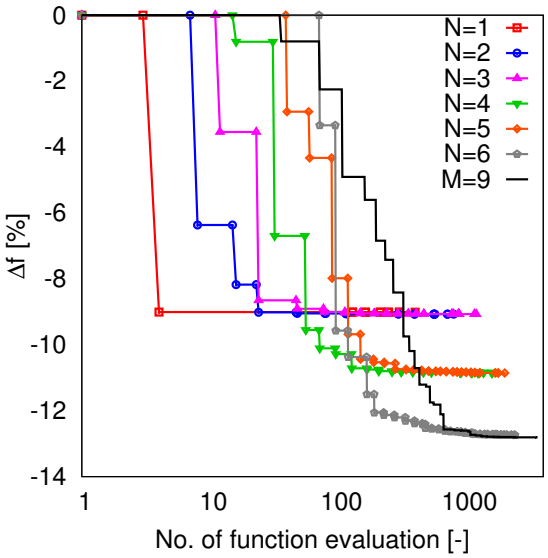


Figure 5.54: Problem VIII, convergence of LS-DF_PSO to the optimal solution, using N-dimensional spaces with $N = 1,\dots,6$

The convergence of LS-DF_PSO is shown in Fig. 5.54, using N-dimensional design space with $N = 1,\dots,6$. The best solution is found using $N = 6$ and provides an objective function reduction close to 13%. Moreover Fig. 5.54 shows the optimization convergence of the original nine-dimensional design space, which reaches about 13% reduction of the objective function. The optimal values for the normalized design variables are shown in Fig. 5.55. The optimal shapes and their associated normalized wave elevation (η/LBP) patterns are shown and compared to the original in Figs. 5.56 and 5.57, respectively. Table 5.19 summarizes the optimization results achieved in the current study.

Finally, the objective function reduction versus the geometric variance retained by each reduced-dimensionality design-space is depicted in Fig. 5.58, showing the significant impact of the geometric variance on the optimization achievements.

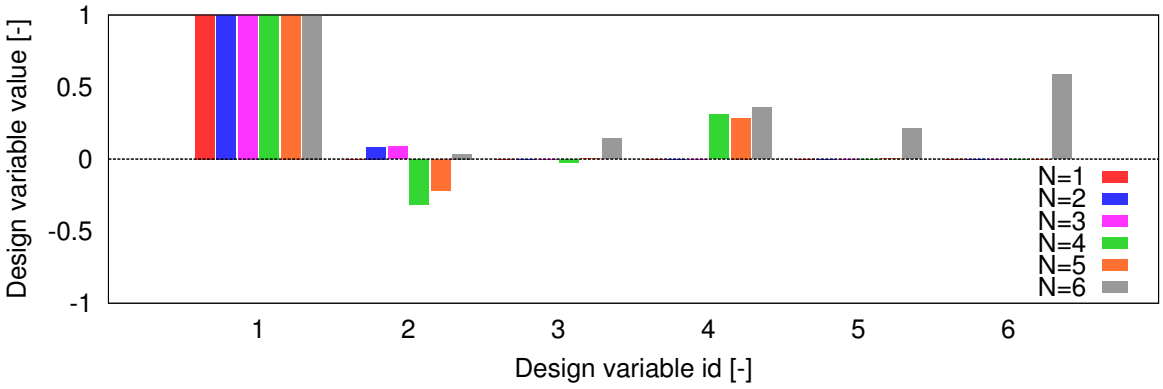
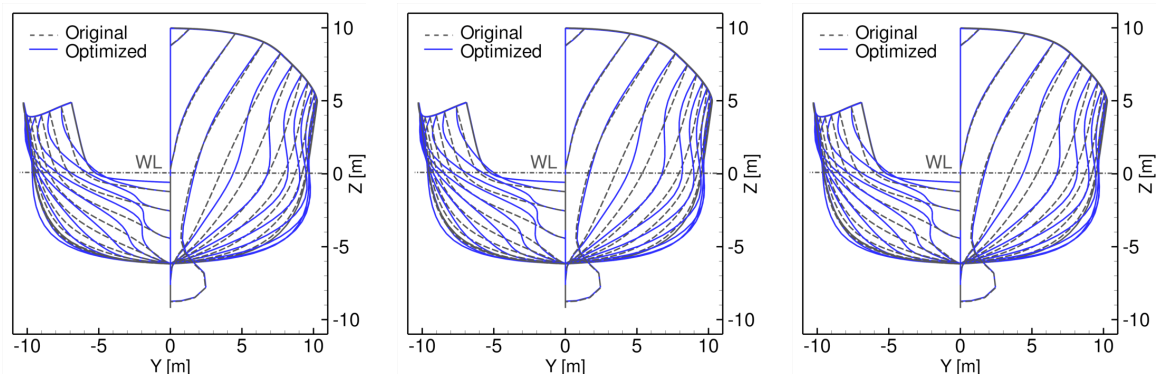
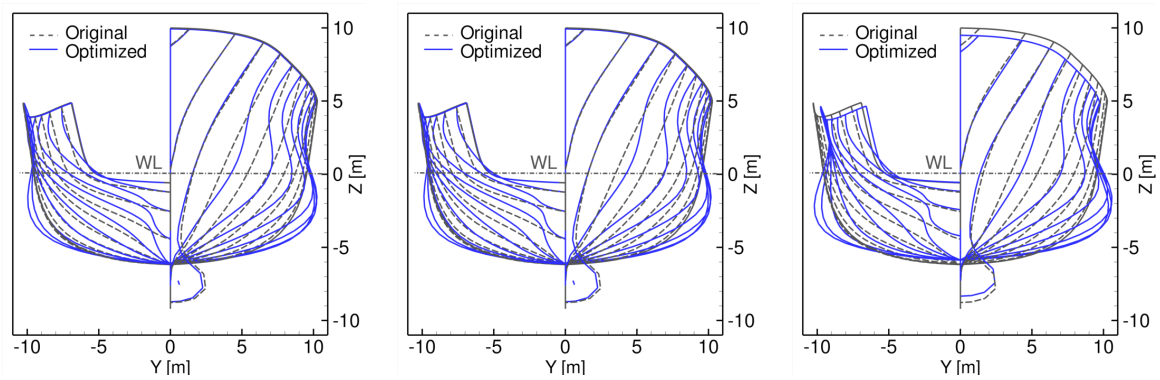


Figure 5.55: Problem VIII, optimal design variables

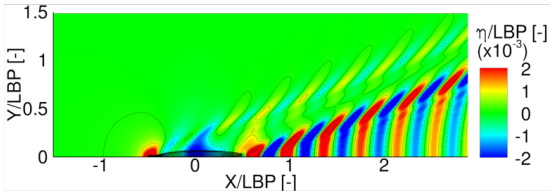


(a) $N = 1$ (b) $N = 2$ (c) $N = 3$

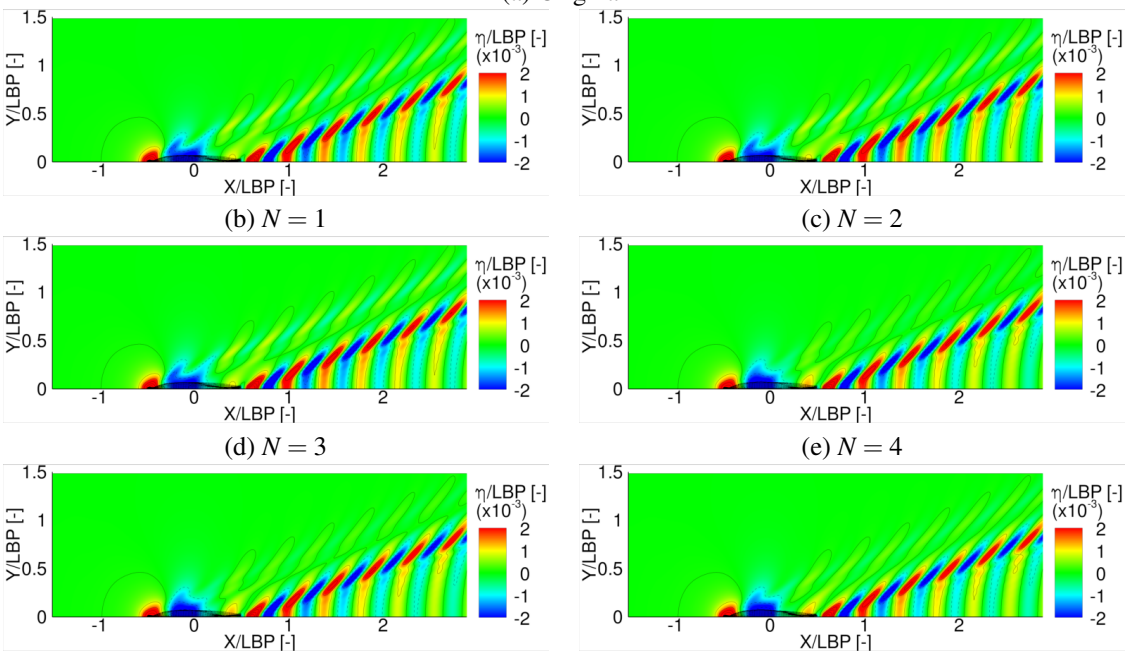


(a) $N = 4$ (b) $N = 5$ (c) $N = 6$

Figure 5.56: Problem VIII, optimized hulls compared to the original



(a) Original



(b) $N = 1$ (c) $N = 2$

(d) $N = 3$ (e) $N = 4$

(f) $N = 5$ (g) $N = 6$

Figure 5.57: Problem VIII, optimized wave patterns compared to the original

Table 5.19: Problem VIII, summary of the DTMB 5415 optimization results

N	$R_T(\times 10^5)$ [N]	$\Delta R_T\%$	$\nabla(\times 10^3)$ [tonnes]	$\Delta\nabla\%$	$\Delta f\%$
1	2.897	-9.017	8.449	-0.005	-9.011
2	2.895	-9.075	8.449	-0.005	-9.070
3	2.895	-9.076	8.449	-0.005	-9.071
4	2.839	-10.85	8.449	-0.002	-10.84
5	2.838	-10.86	8.449	-0.002	-10.86
6	2.773	-12.92	8.433	-0.202	-12.74

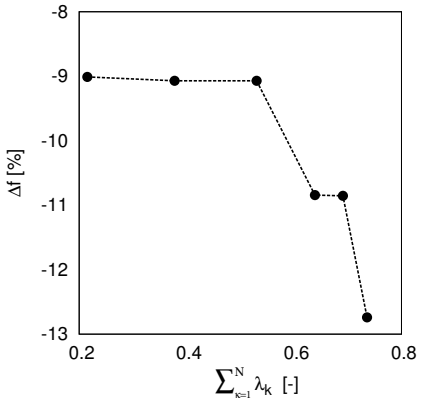
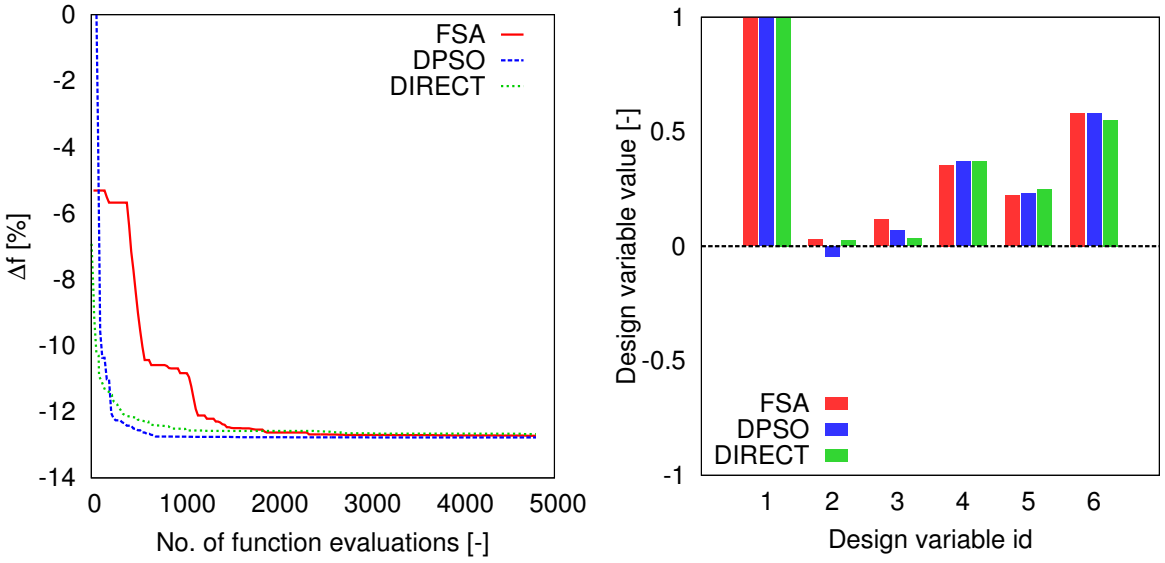


Figure 5.58: Problem VIII, optimization achievement versus geometric variance retained

Problem IX

The preliminary sensitivity analysis for each design variables is the same of Problem VIII (see Fig. 5.53). Changes in f reveal a possible reduction of the objective function close to 10%.

The optimizations are performed with both FSA, using the guideline suggested in Tab. 5.9, DPSO [44], and DIRECT. Optimization results are summarized in Tab. 5.20. FSA, DPSO and DIRECT reaches an objective function reduction close 13%. Specifically, FSA, DPSO, and DIRECT have a similar objective function reduction, even if DPSO and DIRECT shows a faster convergence to the global minimum, as depicted in Fig. 5.59a. The final configurations found by the three algorithms are very close in terms of design variables (see Fig. 5.59b) and show almost the same differences compared to the original design, in terms of wave elevation pattern (see Fig. 5.60a, 5.60c, and 5.60e) and hull sections (see Fig. 5.60b, 5.60d, and 5.60f).



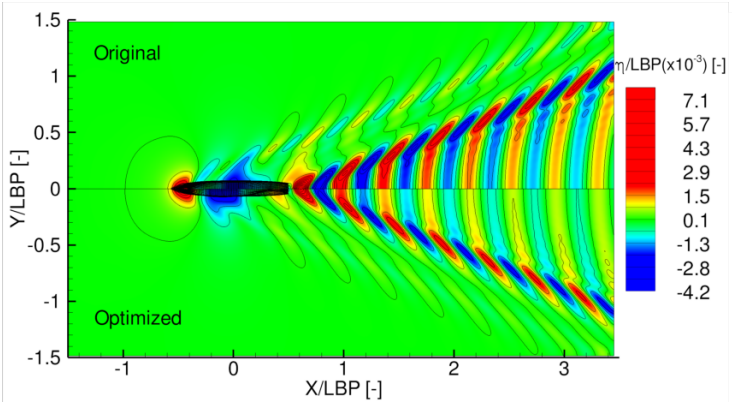
(a) Convergence history

(b) Optimum design variable

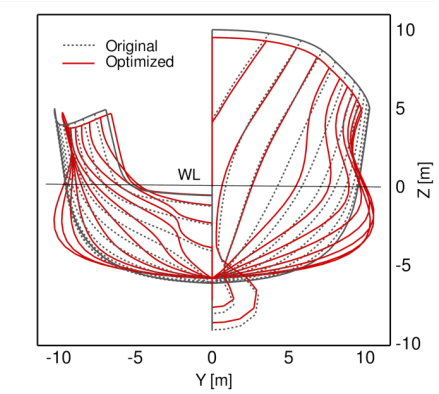
Figure 5.59: Problem IX, optimization results

Table 5.20: Problem IX, SBD optimization results

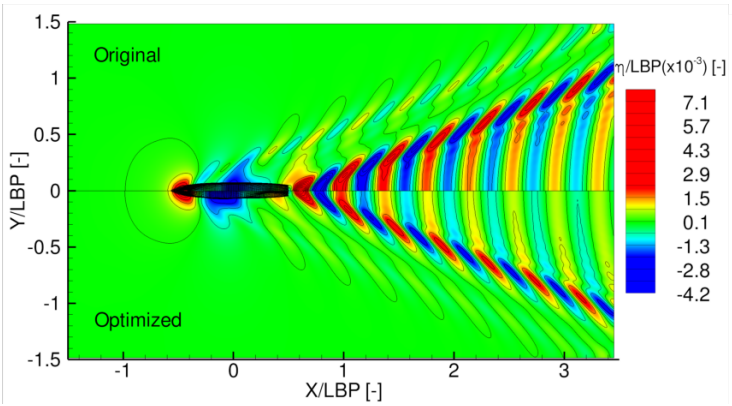
Algorithm	x_1	x_2	x_3	x_4	x_5	x_6	$\Delta f\%$
FSA	0.999	0.030	0.116	0.353	0.221	0.581	-12.73
DPSO	1.000	-0.044	0.069	0.370	0.229	0.579	-12.78
DIRECT	0.996	0.025	0.033	0.370	0.247	0.551	-12.68



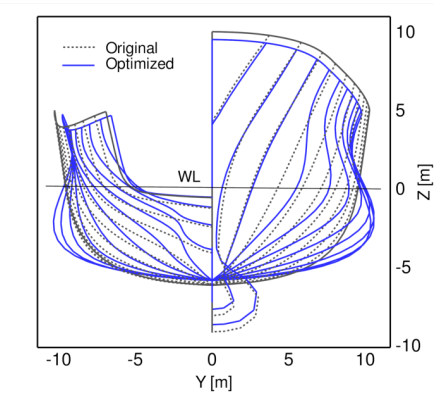
(a) FSA, wave field



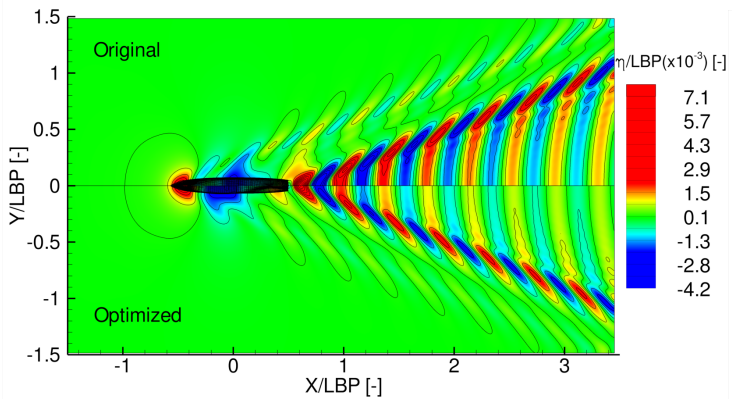
(b) FSA, hull sections



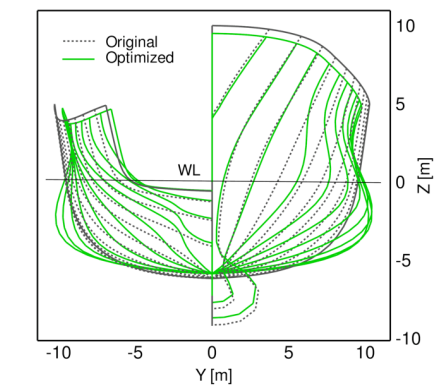
(c) DPSO, wave field



(d) DPSO, hull sections



(e) DIRECT, wave field



(f) DIRECT, hull sections

Figure 5.60: Problem IX, optimization results compared to the original

Problem X

A preliminary sensitivity analysis for each design variables is shown in Fig. 5.61, showing the variability of the objective function ($\Delta f\%$). Changes in f are found significant, revealing a reduction of the objective function close to 15%.

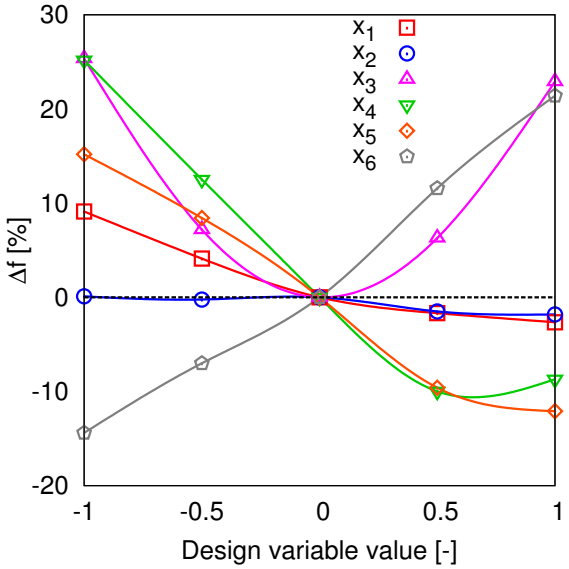
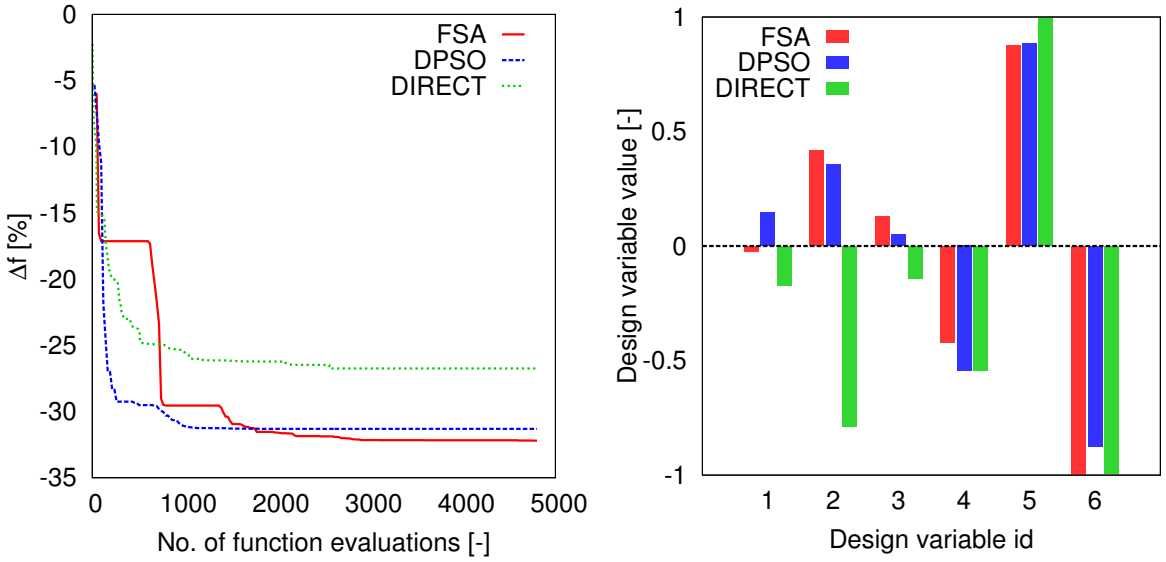


Figure 5.61: Problem X, sensitivity analysis results

The optimizations are performed with both FSA, using the guideline suggested in Tab. 5.9, DPSO [44], and DIRECT. Optimization results are summarized in Tab. 5.21. FSA, DPSO, and DIRECT reach an objective function reduction close 32%, 31% and 27%, respectively.



(a) Convergence history

(b) Optimum design variable

Figure 5.62: Problem X, optimization results

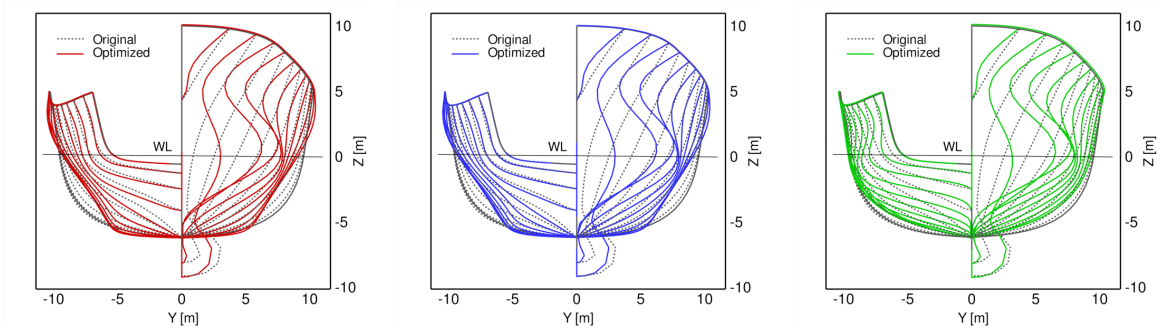
Specifically, results show a greater objective function improvement by FSA, compared with DPSO and specially DIRECT (see Fig. 5.62a), corresponding to different optimal design variables (see Fig. 5.62b).

The optimal design variables x_1, x_2, and x_3 goes in different direction, whereas x_4, x_5, and x_6 goes in the same direction for both FSA, DPSO, and DIRECT, and this aspect reflect the small differences between FSA, DPSO

and DIRECT optimal design, in terms of hull section (see Fig. 5.63) and response amplitude operator of heave, pitch and roll motions (see Fig. 5.64). In particular, it can be seen how the DIRECT optimal design has a slightly different behaviour, compared to FSA and DPSO, in the roll motion RAO (see Fig. 5.64c).

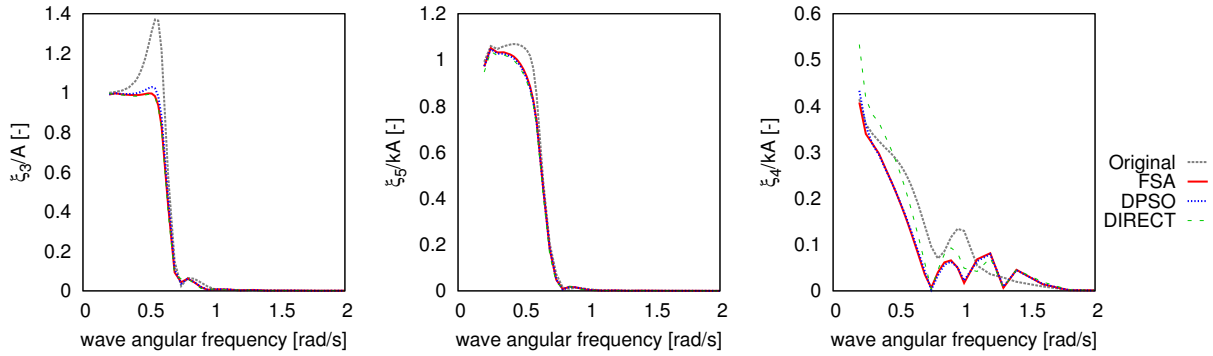
Table 5.21: Problem X, SBD optimization results

Algorithm	x_1	x_2	x_3	x_4	x_5	x_6	$\Delta f\%$
FSA	-0.022	0.417	0.127	-0.421	0.874	-0.997	-32.20
DPSO	0.146	0.356	0.049	-0.545	0.884	-0.873	-31.31
DIRECT	-0.173	-0.787	-0.140	-0.543	0.993	-0.993	-26.65



(a) FSA (b) DPSO (c) DIRECT

Figure 5.63: Problem X, optimized hull sections compared to the original



(a) 30 kn, head wave (b) 30 kn, head wave (c) 18 kn, stern wave

Figure 5.64: Problem X, response amplitude operator for (a) heave, (b) pitch, and (c) roll motion

Chapter 6

Conclusions and future work

"Tut, tut, child!" said the Duchess."Everything's got a moral, if only you can find it."

— Lewis Carroll, *Through the Looking-Glass*

The present work have introduced, assessed, and applied six deterministic derivative-free global and global/local optimization algorithm, along with three shape modification techniques, for an efficient and effective use in the SBDO context. The focus has been placed mainly on the optimization algorithms, describing the features and the convenience of using global deterministic derivative-free approaches (see Chapter 1), and the possible improvements of the local hybridization techniques. DPSO, LS-DF_PSO, DIRECT, DIRMIN, DIRMIN-2, and FSA have been tested on a benchmark of seventy-three analytical function, in order to assess the algorithms performance and define guidelines for an effective and efficient use in a computational framework characterized by limited resources. The most promising algorithms setups have been discussed, identified, and successfully applied to ten ship SBDO problems, pertaining the hull-form optimization of the high-speed Delft catamaran (problems I÷IV, see Sec. 4.2.1) and an USS Arleigh Burke-class destroyer ship, namely the DTMB 5415 model, an early and open-to-public version of the DDG-51 (problems V÷X, see Sec. 4.2.2). The naval engineering optimization problems are based on low- and high-fidelity solvers (see Chapter 3). The main outcomes are recalled in the following.

The parametric analysis (Sec. 5.1.1) for S-DPSO and A-DPSO has been performed varying the number of particles, the initialization of the swarm, the set of coefficients and the wall-type approach for the box constraints. The assessment is based on sixty analytical test functions (with dimensionality from two to twenty) and three different absolute performance criteria. All possible combinations of DPSO parameters led to 420 optimizations for each function. The particles initialization has been found the most significant parameter for the DPSO performance, especially for a number of design variables $N \geq 10$ and low budgets of function evaluations. Conversely, the coefficient set and the wall-type approach have been found having a little influence on the DPSO performance, compared to the other parameters. For problem with less then 10 variables, the suggested S-DPSO and A-DPSO setups coincide except for the wall-type approach: (a) number of particles N_p equal to 4 times the number of design variables; (b) particles initialization including HSS distribution on domain and bounds with non-null velocity; (c) set of coefficient proposed in [110], i.e., $\chi = 0.721$, $c_1 = c_2 = 1.655$; (d) semi-elastic wall-type approach, for S-DPSO and inelastic or semi-elastic wall-type approach for A-DPSO. For problems with more then 10 design variables, the suggested setups differ in the set of coefficients: (a) number of particles N_p equal to 16 times the number of design variables; (b) particles initialization including HSS distribution on domain with non-null velocity; (c) set of coefficient proposed in [59] for S-DPSO, i.e., $\chi = 0.600$, $c_1 = c_2 = 1.700$, and in [110] for A-DPSO; (d) semi-elastic wall-type approach. The performance of the suggested guideline for the test functions has been found always very close or coincident to the best setup among all 420 available. The suggested S-DPSO and A-DPSO guidelines have been proven to perform well for the $N = 4$ and $N = 6$ ship SBDO problems (namely problem I and II). A-DPSO has been found having equivalent performance (in terms of number of objective function evaluations and objective reduction) of the S-DPSO, providing opportunities for exploitation of parallel architectures in high performance computing systems. Finally, the semi-elastic wall-type approach for box constraints should be preferred to the inelastic wall-type, in order to avoid an early halt of the swarm particles dynamics. Moreover, the synchronous implementation of DPSO has been used as reference for problems III, IV, VI, VII, IX, and X.

The globally convergent modifications of DPSO iteration, for the solution of the unconstrained global optimization problem, have been detailed in Sec. 1.2. Under mild assumptions, it has been proven that at least a subsequence of the iterates generated by the modified DPSO (LS-DF_PSO), converges to a stationary point, which is possibly a minimum point. The LS-DF_PSO is also among the first attempts to couple DPSO with line search-

based derivative-free schemes. On the basis of the present results (see Sec. 5.1.2), a fruitful coupling of DPSO with an iterative globally convergent derivative-free method, should yield a compromise, between the fast progress of DPSO (global search) in the early iterations, and the capability to exploit (local search) the objective function. In particular the results on the Ronsebronck function and problems III, IV, VI, and VII, confirm that LS-DF_PSO is more effective (and to a great extent equally efficient) than DPSO. Indeed, LS-DF_PSO is able to achieve better solution, and provides stationarity properties at the associated optimal points.

Two global/local hybrid algorithms, namely DIRMIN and DIRMIN-2, have been presented and tested on seventy-two analytical test functions, separately considering problems with $N \leq 6$ and $N > 6$ variables (Sec. 5.1.3). The two algorithms differ in the derivative-free local search management. In particular, while DIRMIN executes the local search starting from the centers of all the potentially optimal hyper-rectangles, DIRMIN-2 performs a single local minimization starting from the best point produced by dividing the potentially optimal hyper-rectangles. Twenty-five different setups of the algorithms have been investigated, varying the local search activation trigger γ and the local search tolerance β. Data and performance profiles, along with absolute evaluation metrics, have been used to identify the most promising setup for both DIRMIN and DIRMIN-2. The analytical test problem results have revealed that, for both low and high dimensional problems, DIRMIN and DIRMIN-2 are mainly affected by the local search activation trigger. Specifically, the numerical experiments suggest starting the derivative-free local searches at the beginning of the optimization procedures. Furthermore, the two hybrid algorithms are found more effective and efficient than the original DIRECT algorithm, with beneficial effects on the overall computational cost, in view of SBDO. Regarding the DTMB 5415 hull-form optimization (problem V), DIRMIN and DIRMIN-2 have similar performances and show a significantly faster convergence than the original DIRECT algorithm. Moreover, DIRECT and DIRMIN-2 have been applied on problems VI and VII, showing the efficiency and effectiveness of the local hybridization.

Furthermore, a novel fish shoal algorithm (FSA) for deterministic global derivative-free optimization has been presented (see Sec. 1.5). FSA is based on a simplified social model of a fish shoal in search for food, and it is intended for unconstrained single-objective maximization. A parametric analysis has been conducted using sixty analytical test functions and three evaluation metrics, varying the number of individuals, their initialization, the coefficient set controlling the shoal dynamics, and the box constraints method (see Sec. 5.1.4). All possible combinations of FSA implementations led to 486 optimizations for each function. The coefficient q (weight of the attraction forces) and the shoal initialization are the most significant parameters for low budget of function evaluations and $N < 10$, while for $N \geq 10$ the shoal initialization is found always the most significant. The most promising FSA setup has been identified and corresponds to: a number of individuals N_s equal to 4 times the number of design variables; a shoal initialization with a distribution over the whole design variables domain for $N < 10$ and only on the domain boundary for $N \geq 10$; a set of coefficient corresponding to: $\xi = 1.00$, $q = 1.00$ and $p = 8$ for $N < 10$, and to $\xi = 0.10$, $q = 0.10$ and $p = 4$ for $N \geq 10$; and an elastic wall-type approach. FSA has been found slightly better than DPSO and DIRECT for $N < 10$ and always better for $N \geq 10$, in term of effectiveness. A further study on the use of the food-related attraction in FSA has been presented, varying the modeling of the food-related attraction. The use of $\alpha = 0.5$ results in the most efficient and effective modeling for the food-related attraction. The best performing FSA has been applied to two six-dimensional ship SBDO problems (namely problems IX and X), showing comparable results between FSA, DPSO, and DIRECT. In particular, problem X shows the better effectiveness of FSA, compared to DPSO and DIRECT.

About the shape design modification methodologies, FFD, OBFE_2D, and OBFE_3D technique have presented and discussed (Chapter 2), along with the investigation of a KLE-based dimensionality reduction method. The mathematical derivation of the design-space dimensionality reduction has been presented, connected to a global continuous representation of the shape modification vector. The associated structure and breakdown of the geometric variance has been investigated through the eigenvalues and eigenmodes provided by the KLE. The dimensionality reduction for the shape optimization problem is based on the eigenvalues, which represent the geometric variance associated to the corresponding eigenmodes. The reduced-dimensionality design space is defined using the eigenmodes, as new basis functions, and has been applied on problem VIII, where the results have shown the KLE capability of reducing the design-space dimensionality, while retaining a prescribed level of design variability and achieving the same optimization results of the original design space. The formulation presented goes beyond the application and is suitable in all areas where the shape design is a critical issue and involves complex single- and multi-disciplinary simulations.

In conclusion, the present research has firstly assessed the setups of the deterministic derivative-free optimization algorithms (used herein) in the SBDO context, filling the gap of this field; secondly, has introduced a novel global metaheuristic (FSA) capable to compete effectively with two well-known algorithms, as PSO and DIRECT; finally, has shown how global/local hybridization methods, LS-DF_PSO, DIRMIN, and DIRMIN-2, outperform their original global algorithms, DPSO and DIRECT, especially for low budgets of function evaluations. Hybrid algorithms have shown their capability to combine effectively the characteristics of global and local approaches,

resulting in a faster (and computationally less expensive) convergence towards the global minimum. This, along with their derivative-free formulation and implementation, makes the present local hybridization methods a viable and effective option for SBDO, especially when computationally expensive objective functions (such as RANS) are involved.

Future work includes the extension of global/local hybridization methods to multi-objective problems along with metamodel-based optimization by adaptive sampling procedures [26], as well as the local hybridizations of FSA. This last algorithm, originally introduced in this work, will be further extended including the possibility of using two different coefficients for the two attraction forces, as well as dynamically varying mass and damping coefficient. Moreover, since the present FSA formulation is intended as fully informed, a "less informed" formulation could be investigated to improve the FSA performances in terms of efficiency.

Appendix A

Test functions

"Now, here, you see, it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!"

— Lewis Carroll, *Through the Looking Glass*

This appendix provides the analytical formulation used for the test functions of Tab. 4.1.

Ackley function

$$f(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}} - e^{\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)} + 20 + e \tag{A.1}$$

Alpine function

$$f(\mathbf{x}) = \sum_{i=1}^{N} |x_i \sin(x_i) + 0.1x_i| \tag{A.2}$$

Beale function

$$f(\mathbf{x}) = (1.5 + x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2 \tag{A.3}$$

Booth function

$$f(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \tag{A.4}$$

Bukin No.6 function

$$f(\mathbf{x}) = 100\sqrt{|x_2 - 0.01x_1^2|} + 0.01\,|x_1 + 10| \tag{A.5}$$

Colville function

$$f(\mathbf{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_3 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1) \tag{A.6}$$

Cosine Mixture function

$$f(\mathbf{x}) = -\sum_{i=1}^{N}\left[\frac{1}{10}\cos(5\pi x_i) - x_i^2\right] \tag{A.7}$$

Dixon-Price function

$$f(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^{N}\left[i(2x_i^2 - x_{i-1})^2\right] \tag{A.8}$$

Easom function

$$f(\mathbf{x}) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2} \tag{A.9}$$

Exponential function

$$f(\mathbf{x}) = -\exp\left(-\frac{1}{2}\sum_{i=1}^{N} x_i^2\right) \tag{A.10}$$

Freudenstein-Roth function

$$f(\mathbf{x}) = (-13 + x_1 + ((5 - x_2))x_2 - 2)x_2)^2 + (-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2)^2 \tag{A.11}$$

Goldstein-Price function

$$\begin{array}{c} f(\mathbf{x}) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \\ \cdot \left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right] \end{array} \tag{A.12}$$

Griewank function

$$f(\mathbf{x}) = 1 + \sum_{i=1}^{N} \frac{x_i}{4000} - \prod_{i=1}^{N} \cos(x_i/\sqrt{i}) \tag{A.13}$$

Hartman n.3 function

$$f(\mathbf{x}) = -\sum_{i=1}^{4} a_i \exp\left[-\sum_{j=1}^{3} b_{ij}(x_j - d_{ij})^2\right] \tag{A.14}$$

with

$$\mathbf{a} = \begin{Bmatrix} 1.0 \\ 1.2 \\ 3.0 \\ 3.2 \end{Bmatrix} \qquad \mathbf{b} = \begin{Bmatrix} 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \\ 3.0 & 10.0 & 30.0 \\ 0.1 & 10.0 & 35.0 \end{Bmatrix} \qquad \mathbf{d} = \begin{Bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{Bmatrix}$$

Hartman n.6 function

$$f(\mathbf{x}) = -\sum_{i=1}^{4} a_i \exp\left[-\sum_{j=1}^{6} b_{ij}(x_j - d_{ij})^2\right] \tag{A.15}$$

with

$$\begin{array}{c} \mathbf{a} = \begin{Bmatrix} 1.0 \\ 1.2 \\ 3.0 \\ 3.2 \end{Bmatrix} \qquad \mathbf{b} = \begin{Bmatrix} 10.0 & 3.0 & 17.0 & 3.5 & 1.7 & 8.0 \\ 0.05 & 10.0 & 17.0 & 0.1 & 8.0 & 14.0 \\ 3.0 & 3.5 & 1.7 & 10.0 & 17.0 & 8.0 \\ 17.0 & 8.0 & 0.05 & 10.0 & 0.1 & 14.0 \end{Bmatrix} \\ \mathbf{d} = \begin{Bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{Bmatrix} \end{array} \tag{A.16}$$

Levy 5^n function

$$f(\mathbf{x}) = \frac{\pi}{N}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{N-1}\left[(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1}))\right] + (y_N - 1)^2\right\} \tag{A.17}$$

with $y_i = 1 + \frac{1}{4}(x_i - 1)$

Levy 10^n function

$$f(\mathbf{x}) = \frac{\pi}{N}\left\{10\sin^2(\pi x_1) + \sum_{i=1}^{N-1}\left[(x_i - 1)^2(1 + 10\sin^2(\pi x_{i+1}))\right] + (x_N - 1)^2\right\} \tag{A.18}$$

Levy 15^n function

$$f(\mathbf{x}) = \frac{1}{10}\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{N-1}\left[(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1}))\right]\right\} + \frac{1}{10}(x_N - 1)^2\left[1 + \sin^2(2\pi x_N)\right] \tag{A.19}$$

Matyas function

$$f(\mathbf{x}) = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2 \tag{A.20}$$

Multi Modal function

$$f(\mathbf{x}) = \sum_{i=1}^{N} |x_i| \prod_{i=1}^{N} |x_i| \tag{A.21}$$

Powell function

$$f(\mathbf{x}) = \sum_{i=2}^{N/4} \left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} + x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4 \right] \tag{A.22}$$

Quartic function

$$f(\mathbf{x}) = \frac{x_1^4}{4} - \frac{x_1^2}{2} + \frac{x_1}{10} + \frac{x_2^2}{2} \tag{A.23}$$

Rastrigin function

$$f(\mathbf{x}) = 10N + \sum_{i=1}^{N} \left[x_i^2 - 10\cos(2\pi x_i) \right] \tag{A.24}$$

Rosenbrock function

$$f(\mathbf{x}) = (1 - x_1)^2 + 100 \left(x_2 - x_1^2 \right)^2 \tag{A.25}$$

Schaffer n.2 function

$$f(\mathbf{x}) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \tag{A.26}$$

Schaffer n.6 function

$$f(\mathbf{x}) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \tag{A.27}$$

Schwefel function

$$f(\mathbf{x}) = 418.9829N - \sum_{i=1}^{N} \left[x_i \sin\left(\sqrt{|x_i|} \right) \right] \tag{A.28}$$

Shekel n.5 function

$$f(\mathbf{x}) = -\sum_{j=1}^{5} \left[c_j + \sum_{i=1}^{4} (x_i - A_{i,j})^2 \right]^{-1} \tag{A.29}$$

with

$$\mathbf{c} = \frac{1}{10} \begin{bmatrix} 1 & 2 & 2 & 4 & 4 \end{bmatrix}^T \qquad \mathbf{A} = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 \\ 4 & 1 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 3 \\ 4 & 1 & 8 & 6 & 7 \end{bmatrix}^T$$

Shekel n.7 function

$$f(\mathbf{x}) = -\sum_{j=1}^{7} \left[c_j + \sum_{i=1}^{4} (x_i - A_{i,j})^2 \right]^{-1} \tag{A.30}$$

with

$$\mathbf{c} = \frac{1}{10} \begin{bmatrix} 1 & 2 & 2 & 4 & 4 & 6 & 3 \end{bmatrix}^T \qquad \mathbf{A} = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 \\ 4 & 1 & 8 & 6 & 7 & 9 & 5 \\ 4 & 1 & 8 & 6 & 3 & 2 & 3 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 \end{bmatrix}^T$$

Shekel n.10 function

$$f(\mathbf{x}) = -\sum_{j=1}^{10}\left[c_j + \sum_{i=1}^{4}(x_i - A_{i,j})^2\right]^{-1} \tag{A.31}$$

with

$$\mathbf{c} = \frac{1}{10}\begin{bmatrix}1 & 2 & 2 & 4 & 4 & 6 & 3 & 7 & 5 & 5\end{bmatrix}^T \qquad \mathbf{A} = \begin{bmatrix}4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7\\ 4 & 1 & 8 & 6 & 7 & 9 & 5 & 1 & 2 & 3.6\\ 4 & 1 & 8 & 6 & 3 & 2 & 3 & 8 & 6 & 7\\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6\end{bmatrix}^T$$

Shubert penalty 1 function

$$f(\mathbf{x}) = \prod_{i=1}^{2}\left(\sum_{j=1}^{5} j\cos((j+1)x_i + j)\right) + \left((x_1 + 1.42513)^2 + (x_2 + 0.80032)^2\right)/2 \tag{A.32}$$

Shubert penalty 2 function

$$f(\mathbf{x}) = \prod_{i=1}^{2}\left(\sum_{j=1}^{5} j\cos((j+1)x_i + j)\right) + (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2 \tag{A.33}$$

Six-humps Camel Back function

$$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{1}{3}x_1^4\right)x_1^2 + x_1x_2 + \left(4x_2^2 - 4\right)x_2^2 \tag{A.34}$$

Sphere function

$$f(\mathbf{x}) = \sum_{i=1}^{N} x_i^2 \tag{A.35}$$

Styblinski-Tang function

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{N}\left(x_i^4 - 16x_i^2 + 5x_i\right) \tag{A.36}$$

Test Tube Holder function

$$f(\mathbf{x}) = -4\left|e^{\left|\cos\left(\frac{1}{200}x_1^2 + \frac{1}{200}x_2^2\right)\right|}\sin(x_1)\cos(x_2)\right| \tag{A.37}$$

Three-humps Camel Back function

$$f(\mathbf{x}) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 + x_1x_2 + x_2^2 \tag{A.38}$$

Treccani function

$$f(\mathbf{x}) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2 \tag{A.39}$$

Tripod function

$$\begin{aligned} f(\mathbf{x}) = \; & \tfrac{1-sign(x_2)}{2}(|x_1| + |x_2 + 50|) + \\ & + \tfrac{1+sign(x_2)}{2}\tfrac{1-sign(x_1)}{2}(1 + |x_1 + 50| + |x_2 - 50|) + \\ & + \tfrac{1+sign(x_2)}{2}\tfrac{1+sign(x_1)}{2}(2 + |x_1 - 50| + |x_2 - 50|) \end{aligned} \tag{A.40}$$

with

$$sign(x_i) = \begin{cases} -1, & \text{if } x_i \leq 0 \\ 1, & \text{else} \end{cases}$$

Appendix B

Metamodels

"Imagination is the only weapon in the war against reality."

— Lewis Carroll, *Alice in Wonderland*

The polyharmonic spline is a linear combination of polyharmonic radial basis functions (RBFs) [26] denoted by ϕ:

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(||\mathbf{x} - \mathbf{c}_i||) + \mathbf{v}^T \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \tag{B.1}$$

where $\mathbf{x} = [x_1 \dots x_d]^T$ is a real-valued vector of d independent variables, $\mathbf{c}_i = [c_{i,1} \dots c_{i,d}]^T$ are the N vectors of the same size as $\mathbf{x}$ that the curve/surface must interpolate, $\mathbf{w} = [w_1 \dots w_N]^T$ are the N weights of the RBFs, and $\mathbf{v} = [v_1 \dots v_{d+1}]^T$ are the $d+1$ weights of the polynomial. The polynomial with the coefficients $\mathbf{v}$ improves fitting accuracy for polyharmonic smoothing splines and also improves extrapolation away from the centers $\mathbf{c}_i$.

The polyharmonic RBFs are of the form:

$$\phi(r) = \begin{cases} r^k & \text{with} \quad k = 1,3,5,\dots \\ r^k \ln(r) & \text{with} \quad k = 2,4,6,\dots \end{cases} \tag{B.2}$$

where $r = ||\mathbf{x} - \mathbf{c}_i||$.

The weight vector $\mathbf{w} = \{w_i\}$ and $\mathbf{v} = \{v_j\}$ are the solution of the linear system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \tag{B.3}$$

where

$$A_{i,j} = \phi(||\mathbf{c}_i - \mathbf{c}_j||), \quad \mathbf{B} = \begin{bmatrix} 1 & \dots & 1 \\ \mathbf{c}_1 & \dots & \mathbf{c}_N \end{bmatrix}^T, \quad \mathbf{f} = [f_1, \dots, f_N]^T \tag{B.4}$$

List of Figures

List of Tables

List of Algorithms

List of Publications

Chapters published in Book

1. Serani A., Diez M., Campana E.F., Fasano G., Peri D., Iemma U., "Globally Convergent Hybridization of Particle Swarm Optimization Using Line Search-Based Derivative-Free Techniques" In: Recent Advances in Swarm Intelligence and Evolutionary Computation, Studies in Computational Intelligence, Vol. 585, Yang, Xin-She (Ed.), Springer, 2015.

2. Diez M., Serani A., Leotardi C., Campana E.F., Peri D., Iemma U., Fasano G., Giove S., "A proposal of PSO particles initialization for costly unconstrained optimization problems: ORTHOinit," 5th International Conference, ICSI 2014, Hefei, China, October 17-20, 2014, Proceedings, Part I. In: Advances in Swarm Intelligence, Lecture Notes in Computer Science Volume 8794, 2014, pp 126-133.

Papers published in Peer-reviewed International Journals

1. Serani A., Fasano G., Liuzzi G., Lucidi S., Iemma U., Campana E.F., Diez M., "Ship hydrodynamic optimization by local hybridization of deterministic derivative-free global algorithms," In press. Applied Ocean Research, Elsevier 2016.

2. Leotardi C., Serani A., Iemma U., Campana E.F., Diez M., "A variable-accuracy metamodel-based architecture for global MDO under uncertainty," In press. Structural Multidisciplinary Optimization, Springer, 2016.

3. Campana E.F., Diez M., Iemma U., Liuzzi G., Lucidi S., Rinaldi F., Serani A., "Derivative-free global ship design optimization using global/local hybridization of the DIRECT algorithm," Optimization and Engineering 17(1) 127-156, 2015.

Papers published in International Conference Proceedings

1. Diez M., Serani A., Stern F., Campana E.F., "Combined Geometry and Physics Based Method for Design-space Dimensionality Reduction in Hydrodynamic Shape Optimization", 31st Symposium on Naval Hydrodynamics, Monterey, CA, USA, 11-16 September 2016.

2. Iemma U., Marchese V., Serani A., Palma G., "Optimal design of phononic crystals using higher order boundary elements and a swarm optimization scheme," 23rd International Congress on Sound & Vibration, ICSV23, Athens, Greece, 10-14 July 2016.

3. Diez M., Serani A., Leotardi C., Campana E.F., Fasano G., Gusso R., "Dense Orthogonal Initialization for Deterministic PSO: ORTHOinit+," 7th International Conference, ICSI 2016, Bali, Indonesia, June 25-30, 2016, Proceedings, In: Lecture Notes in Computer Science.

4. Diez M., Serani A., Campana E.F., Stern F., "Design Space Dimensionality Reduction for Single- and Multi-Disciplinary Shape Optimization", AIAA/ISSMO Multidisciplinary Analysis and Optimization (MA&O), AVIATION 2016, Washington D.C., USA, 13-17 June 2016.

5. Serani A., Campana E.F., Diez M., Stern F., "A multi-objective optimization of DTMB 5415: effects of potential flow formulation and correlation with RANS," 15th Conference on Computer Applications and Information Technology in the Maritime Industries, COMPIT 2016, Lecce, Italy, 9-11 May 2016.

6. Diez M., Volpi S., Serani A., Stern F., Campana E.F., "Simulation-based Design Optimization by Sequential Multi-criterion Adaptive Sampling and Dynamic Radial Basis Functions," EUROGEN 2015 - International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, Glasgow, UK, September 2015.

7. Diez, M., Serani, A., Campana, E.F., Goren, O., Sarioz, K., Danisman, D., Grigoropoulos, G., Aloniati, E., Visonneau, M., Queutey, P., Stern, F., "Multi-objective hydrodynamic optimization of the DTMB 5415 for resistance and seakeeping," In: Proceedings of the 13th international conference on fast sea transportation, FAST 2015, Washington, DC, September 2015.

8. Serani A., Fasano G., Liuzzi G., Lucidi S., Iemma U., Campana E.F., Diez M., "Derivative-free global design optimization in ship hydrodynamics by local hybridization," 14th International Conference on Computer Applications and Information Technology in the Maritime Industries, COMPIT 2015, Ulrichshusen, Germany, May 2015.

9. Diez M., Serani A., Iemma U., Campana E.F., "A fish shoal algorithm for global derivative-free simulation-based ship design optimization," 17th Numerical Towing Tank Symposium - NuTTS 2014, Marstrand, Sweden, September 2014.

10. Pellegrini R., Serani A., Diez M., Iemma U., Campana E.F., "Multi-objective extensions of the deterministic particle swarm algorithm for RBRDO in ship design: a parametric study," 17th Numerical Towing Tank Symposium - NuTTS 2014, Marstrand, Sweden, September 2014.

11. Pellegrini R., Campana E.F., Diez M., Serani A., Rinaldi F., Fasano G., Iemma U., Lucidi S., Liuzzi G., Stern F., "Application of derivative-free multi-objective algorithms to reliability-based robust design optimization of a high-speed catamaran in real ocean environment," 4th International Conference on Engineering Optimization - EngOpt 2014, Lisbon, Portugal, September 2014.

12. Leotardi C., Diez M., Serani A., Iemma U., Campana E.F., "Efficient simulation-based design optimization for fluid-structure interaction problems affected by uncertainty," (extended abstract only) 17th U.S. National Congress on Theoretical and Applied Mechanics, Michigan State University, June 2014.

13. Leotardi C., Diez M., Serani A., Iemma U., Campana E.F., "A framework for efficient simulation-based multidisciplinary robust design optimization with application to a keel fin of a racing sailboat," OPT-i 2014, 1st International Conference on Engineering and Applied Sciences Optimization, Kos Island, Greece, June 2014.

14. Serani A., Diez M., Leotardi C., Peri D., Fasano G., Iemma U., Campana E.F., "On the use of synchronous and asynchronous single-objective deterministic particle swarm optimization in ship design problems," OPT-i 2014, 1st International Conference on Engineering and Applied Sciences Optimization, Kos Island, Greece, June 2014.

Papers submitted to Peer-reviewed International Journals

1. Serani A., Leotardi C., Fasano G., Iemma U., Campana E.F., Diez M., "Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems," submitted to Applied Soft Computing, 13 November 2015. Under review.

Papers submitted to International Conferences

1. Serani A. Diez M., "On the Food-related Attraction in the Fish Shoal Algorithm," Submitted to ANTS 2016, Tenth International Conference on Swarm Intelligence, Brussels, Belgium, September 7-9, 2016.

Papers in preparation for Peer-reviewed International Journals

1. Serani A., Diez M., "A novel Fish Shoal Algorithm for deterministic global derivative-free simulation-based design optimization," in preparation for Journal of Global Optimization, Springer.

2. Pellegrini R., Serani A., Iemma U., Campana E.F., Diez M., "Empirical study of deterministic particle swarm for multi-objective problems in ship design," in preparation for Swarm Intelligence, Springer.

Bibliography

[1] J. Sobieszczanski-Sobieski, R. T. Haftka, Multidisciplinary aerospace design optimization: survey of recent developments, Structural optimization 14 (1) (1997) 1–23.

[2] N. M. Alexandrov, R. M. Lewis, Analytical and computational aspects of collaborative optimization for multidisciplinary design, AIAA Journal 40 (2) (2002) 301–309.

[3] M. Diez, U. Iemma, Multidisciplinary conceptual design optimization of aircraft using a sound-matching-based objective function, Engineering Optimization 44 (5) (2012) 591–612.

[4] M. Kotinis, A. Kulkarni, Multi-objective shape optimization of transonic airfoil sections using swarm intelligence and surrogate models, Structural and Multidisciplinary Optimization 45 (5) (2012) 747–758.

[5] C. Schillings, S. Schmidt, V. Schulz, Efficient shape optimization for certain and uncertain aerodynamic design, Computers & Fluids 46 (1) (2011) 78–87.

[6] S. Kodiyalam, J. Sobieszczanski-Sobieski, Multidisciplinary design optimisation-some formal methods, framework requirements, and application to vehicle design, International Journal of Vehicle Design 25 (1) (2001) 3–22.

[7] F. Duddeck, Multidisciplinary optimization of car bodies, Structural and Multidisciplinary Optimization 35 (4) (2008) 375–389.

[8] M. Hojjat, E. Stavropoulou, K.-U. Bletzinger, The vertex morphing method for node-based shape optimization, Comput. Methods Appl. Mech. Engrg. 268 (2014) 494–513.

[9] H. Wang, G. Lia, E. Lib, Time-based metamodeling technique for vehicle crashworthiness optimization, Comput. Methods Appl. Mech. Engrg. 199 (2010) 2497–2509.

[10] M. Papadrakakis, Y. Tsompanakis, N. D. Lagaros, Structural shape optimization using evolution strategies, Engineering Optimization 31 (4) (1999) 515–540.

[11] P. Geyer, Component-oriented decomposition for multidisciplinary design optimization in building design, Advanced Engineering Informatics 23 (1) (2009) 12 – 31.

[12] E. F. Campana, D. Peri, Y. Tahara, F. Stern, Shape optimization in ship hydrodynamics using computational fluid dynamics, Computer Methods in Applied Mechanics and Engineering 196 (1-3) (2006) 634–651.

[13] A. Papanikolaou, Holistic ship design optimization, Computer-Aided Design 42 (11) (2010) 1028–1044.

[14] M. Diez, D. Peri, G. Fasano, E. F. Campana, Hydroelastic optimization of a keel fin of a sailing boat: a multidisciplinary robust formulation for ship design, Structural and Multidisciplinary Optimization 46 (4) (2012) 613–625.

[15] M. Kandasamy, D. Peri, Y. Tahara, W. Wilson, M. Miozzi, S. Georgiev, E. Milanov, E. F. Campana, F. Stern, Simulation based design optimization of waterjet propelled Delft catamaran, International Shipbuilding Progress 60 (1) (2013) 277–308.

[16] M. Diez, E. F. Campana, F. Stern, Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion, Computer Methods in Applied Mechanics and Engineering 283 (2015) 1525–1544.

[17] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82.

[18] A. Griewank, Evaluating Derivatives, SIAM Frontieres in applied mathematics, Philadelphia, 2000.

[19] J. Nocedal, S. J. Wright, Evaluating Derivatives, 2nd Edition, Operations Research and Financial Engineering, Springer, New York, 2006.

[20] G. Fasano, S. Lucidi, A nonmonotone truncated newton–krylov method exploiting negative curvature directions, for large scale unconstrained optimization, Optimization Letters 3 (4) (2009) 521–535.

[21] T. G. Kolda, R. M. Lewis, V. Torczon, Optimization by direct search: New perspectives on some classical and modern methods, SIAM review 45 (3) (2003) 385–482.

[22] E. F. Campana, G. Liuzzi, S. Lucidi, D. Peri, V. Piccialli, A. Pinto, New global optimization methods for ship design problems, Optimization and Engineering 10 (4) (2009) 533–555.

[23] A. Serani, M. Diez, E. F. Campana, G. Fasano, D. Peri, U. Iemma, Globally convergent hybridization of particle swarm optimization using line search-based derivative-free techniques, in: X.-S. Yang (Ed.), Recent Advances in Swarm Intelligence and Evolutionary Computation, Vol. 585 of Studies in Computational Intelligence, Springer International Publishing, 2015, pp. 25–47.

[24] E. F. Campana, M. Diez, U. Iemma, G. Liuzzi, S. Lucidi, F. Rinaldi, A. Serani, Derivative-free global ship design optimization using global/local hybridization of the DIRECT algorithm, Optimization and Engineering 17 (1) (2015) 127–156.

[25] A. Borzì, V. Schulz, C. Schillings, G. von Winckel, On the treatment of distributed uncertainties in pde-constrained optimization, GAMM-Mitteilungen 33 (2) (2010) 230–246.

[26] S. Volpi, M. Diez, N. Gaul, H. Song, U. Iemma, K. K. Choi, E. F. Campana, F. Stern, Development and validation of a dynamic metamodel based on stochastic radial basis functions and uncertainty quantification, Structural and Multidisciplinary Optimization 51 (2) (2015) 347–368.

[27] M. Diez, S. Volpi, A. Serani, F. Stern, E. F. Campana, Simulation-based design optimization by sequential multi-criterion adaptive sampling and dynamic radial basis functions, in: EUROGEN 2015, International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, Glasgow, UK, September, 2015.

[28] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[29] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies, Journal of statistical physics 34 (5-6) (1984) 975–986.

[30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the Fourth IEEE Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.

[31] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics–Part B 26 (1) (1996) 29–41.

[32] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, Evolutionary Computation, IEEE Transactions on 6 (2) (2002) 182–197.

[33] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intelligence 1 (1) (2007) 33–57.

[34] M. Dorigo, T. Stützle, Handbook of Metaheuristics, Springer US, Boston, MA, 2010, Ch. Ant Colony Optimization: Overview and Recent Advances, pp. 227–263.

[35] X. L. Li, Z. J. Shao, J. X. Qian, An optimizing method based on autonomous animats: Fish-swarm algorithm, System Engineering Theory and Practice 22 (2003) 32–38.

[36] C. Audet, J. E. Dennis, Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM Journal on Optimization 17 (1) (2006) 188–217.

[37] C. Audet, V. Béchard, S. Le Digabel, Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search, Journal of Global Optimization 41 (2) (2008) 299–318.

[38] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[39] X.-S. Yang, S. Deb, Cuckoo search via levy flights, in: Proceedings of World Congress on Nature and Biologically Inspired Computing (NaBic 2009), Coimbatore, USA, 2009, pp. 210–214.

[40] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: J. Gonzlez, D. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Vol. 284 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2010, pp. 65–74.

[41] J. A. Nelder, R. Mead, A simplex method for function minimization, The Computer Journal 7 (1965) 308–313.

[42] X.-S. Yang, Optimization and metaheuristic algorithms in engineering, in: X.-S. Yang, A. H. Gandomi, S. Talatahari, A. H. Alavi (Eds.), Metaheuristics in Water, Geotechnical and Transport Engineering, Elsevier, Oxford, 2013, pp. 1 – 23.

[43] A. Torn, A. Zilinskas, Global Optimization, Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[44] A. Serani, M. Diez, C. Leotardi, D. Peri, G. Fasano, U. Iemma, E. F. Campana, On the use of synchronous and asynchronous single-objective deterministic particle swarm optimization in ship design problems, in: Proceedings of the 1st International Conference in Engeneering and Applied Sciences Optimization, Kos, Greece, June 4-6, 2014.

[45] D. Jones, C. Perttunen, B. Stuckman, Lipschitzian optimization without the Lipschitz constant, Journal of Optimization Theory and Applications 79 (1) (1993) 157–181.

[46] M. Diez, A. Serani, U. Iemma, E. F. Campana, A fish shoal algorithm for global derivative-free simulation-based ship design optimization, in: Proceedings of the 17th Numerical Towing Tank Symposium - NuTTS 2014, Marstrand, Sweden, September, 2014.

[47] X. Chen, M. Diez, M. Kandasamy, Z. Zhang, E. F. Campana, F. Stern, High-fidelity global optimization of shape design by dimensionality reduction, metamodels and deterministic particle swarm, Engineering Optimization 47 (4) (2015) 473–494.

[48] G. Venter, J. Sobieszczanski-Sobieski, A parallel particle swarm optimization algorithm accelerated by asynchronous evaluations, in: Proceedings of the 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 2005.

[49] B. Koh, A. George, R. Haftka, B. Fregly, Parallel asynchronous particle swarm optimization, Int. J. Numer. Meth. Engng 67 (2006) 578–595. doi:10.1002/nme.1646.

[50] Y. Shi, R. Eberhart, Parameter selection in particle swarm optimization, in: Proceedings of Evolutionary Programming VII (EP98), 1998, pp. 591–600.

[51] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, 1998.

[52] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Proc. ICEC, Washington, DC, 1999, pp. 1951–1957.

[53] R. Eberhart, Y. Shi, Comparing inertia weights and constriction factor in particle swarm optimization, in: Congress on Evolutionary Computing, Vol. 1, 2000, pp. 84–88.

[54] R. Eberhart, Y. Shi, Particle swarm optimization: Developments, applications and resources, in: Proceedings of the 2001 Congress on Evolutionary Computation, 2001, pp. 81–86.

[55] M. Clerc, J. Kennedy, The particle swarm: explosion, stability and convergence in a multi-dimensional complex space, IEEE Trans Evolution. Comput. 6 (1) (2002) 58–73.

[56] M. Diez, A. Serani, C. Leotardi, E. F. Campana, D. Peri, U. Iemma, G. Fasano, S. Giove, Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, October 17-20, 2014, Proceedings, Part I, Springer International Publishing, Cham, 2014, Ch. A Proposal of PSO Particles' Initialization for Costly Unconstrained Optimization Problems: ORTHOinit, pp. 126–133.

[57] W. E. Hart, A stationary point convergence theory for evolutionary algorithms, in: Foundations of Genetic Algorithms 4, Morgan Kaufmann, 1997, pp. 127–134.

[58] W. E. Hart, Evolutionary pattern search algorithms for unconstrained and linearly constrained optimization, IEEE Transactions on Evolutionary Computation 5 (4) (2001) 388–397.

[59] I. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (2003) 317–325.

[60] F. van den Bergh, A. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176 (8) (2006) 937 – 971.

[61] A. Conn, K. Scheinberg, L. Vicente, Introduction to Derivative-Free Optimization, Society for Industrial and Applied Mathematics, 2009.

[62] A. I. F. Vaz, L. N. Vicente, A particle swarm pattern search method for bound constrained global optimization, Journal of Global Optimization 39 (2) (2007) 197–219.

[63] A. I. F. Vaz, L. N. Vicente, Pswarm: A hybrid solver for linearly constrained global derivative-free optimization, Optimization Methods Software 24 (4-5) (2009) 669–685.

[64] D. E. Finkel, Global optimization with the DIRECT algorithm, Ph.D. thesis, North Carolina State University, Raleigh, North Carolina, USA (2005).
URL `http://www.lib.ncsu.edu/resolver/1840.16/4738`

[65] G. Liuzzi, S. Lucidi, V. Piccialli, Exploiting derivative-free local searches in direct-type algorithms for global optimization, Computational Optimization and Applications (2015) 1–27.

[66] R. Paulavičius, Y. D. Sergeyev, D. E. Kvasov, J. Žilinskas, Globally-biased disimpl algorithm for expensive global optimization, Journal of Global Optimization 59 (2) (2014) 545–567. doi:10.1007/s10898-014-0180-4.

[67] G. Liuzzi, S. Lucidi, V. Piccialli, A DIRECT-based approach exploiting local minimizations for the solution of large-scale global optimization problems, Computational Optimization and Applications 45 (2010) 353–375.

[68] D. Lera, Y. D. Sergeyev, Deterministic global optimization using space-filling curves and multiple estimates of lipschitz and hlder constants, Communications in Nonlinear Science and Numerical Simulation 23 (13) (2015) 328 – 342.

[69] D. Di Serafino, G. Liuzzi, V. Piccialli, F. Riccio, G. Toraldo, A modified DIviding RECTangles algorithm for a problem in astrophysics, Journal of Optimization Theory and Applications 151 (1) (2011) 175–190.

[70] Q. Liu, W. Cheng, A modified DIRECT algorithm with bilevel partition, Journal of Global Optimization 60 (3) (2013) 483–499.

[71] Q. Liu, J. Zeng, Global optimization by multilevel partition, Journal of Global Optimization 61 (1) (2014) 47–69.

[72] J. Gablonsky, C. Kelley, A locally-biased form of the DIRECT algorithm, Journal of Global Optimization 21 (1) (2001) 27–37.

[73] D. Jones, DIRECT global optimization, in: C. A. Floudas, P. M. Pardalos (Eds.), Encyclopedia of optimization, Springer, 2009, pp. 725–735.

[74] T. D. Panning, L. T. Watson, N. A. Allen, K. C. Chen, C. A. Shaffer, J. J. Tyson, Deterministic parallel global parameter estimation for a model of the budding yeast cell cycle, Journal of Global Optimization 40 (4) (2008) 719–738.

[75] S. Lucidi, M. Sciandrone, A derivative-free algorithm for bound constrained optimization, Computational Optimization and Applications 21 (2) (2002) 119–142.

[76] A. Serani, M. Diez, A novel fish shoal algorithm for deterministic global derivative-free simulation-based design optimization in ship hydrodynamics, Submitted to Swarm Intelligence.

[77] M. Neshat, G. Sepidnam, M. Sargolzaei, A. N. Toosi, Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications, Artificial Intelligence Review 42 (4) (2012) 965–997.

[78] T. Pitcher, A. Magurran, I. Winfield, Fish in larger shoals find food faster, Behavioral Ecology and Sociobiology 10 (2) (1982) 149–151.

[79] T. Pitcher, Functions of shoaling behaviour in teleosts, in: T. Pitcher (Ed.), The Behaviour of Teleost Fishes, Springer US, 1986, pp. 294–337.

[80] S. G. Reebs, Can a minority of informed leaders determine the foraging movements of a fish shoal?, Animal Behaviour 59 (2) (2000) 403–409.

[81] M. H. Keenleyside, Some aspects of the schooling behaviour of fish, Behaviour 8 (1) (1955) 183–247.

[82] S. Jamshid A., A survey of shape parameterization techniques, Tech. rep. (1999).

[83] D. Sieger, S. Menzel, M. Botsch, New Challenges in Grid Generation and Adaptivity for Scientific Computing, Springer International Publishing, Cham, 2015, Ch. On Shape Deformation Techniques for Simulation-Based Design Optimization, pp. 281–303.

[84] E. F. Campana, D. Peri, Y. Tahara, M. Kandasamy, F. Stern, Numerical optimization methods for ship hydrodynamic design, Transactions - Society of Naval Architects and Marine Engineers Society of Naval Architects and Marine Engineers (2009) 30–67.

[85] D. Sieger, S. Menzel, M. Botsch, RBF morphing techniques for simulation-based design optimization, Engineering with Computers 30 (2) (2013) 161–174.

[86] M. Diez, A. Serani, E. F. Campana, F. Stern, Design space dimensionality reduction for single- and multi-disciplinary shape optimization, in: submitted to AIAA/ISSMO Multidisciplinary Analysis and Optimization (MA&O), AVIATION 2016, Washington D.C., USA, June 13-17, 2016.

[87] S. Menzel, B. Sendhoff, Evolutionary Computation in Practice, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, Ch. Representing the Change - Free Form Deformation for Evolutionary Design Optimization, pp. 63–86.

[88] T. W. Sederberg, S. R. Parry, Free-form deformation of solid geometric models, in: IN PROC. SIGGRAPH 86, 1986, pp. 151–160.

[89] A. H. Barr, Global and local deformations of solid primitives, SIGGRAPH Comput. Graph. 18 (3) (1984) 21–30.

[90] A. Serani, G. Fasano, G. Liuzzi, S. Lucidi, U. Iemma, E. F. Campana, M. Diez, Derivative-free global design optimization in ship hydrodynamics by local hybridization, in: Proceedings of the 14th International Conference on Computer Applications and Information Technology in the Maritime Industries, Ulrichshusen, Germany, May 11-13, 2015, pp. 331–342.

[91] A. Serani, G. Fasano, G. Liuzzi, S. Lucidi, U. Iemma, E. F. Campana, F. Stern, M. Diez, Ship hydrodynamic optimization by local hybridization of deterministic derivative-free global algorithms, Applied Ocean Research.

[92] K.-J. Paik, P. M. Carrica, D. Lee, K. Maki, Strongly coupled fluidstructure interaction method for structural loads on surface ships, Ocean Engineering 36 (1718) (2009) 1346 – 1357.

[93] G. Rozvany, M. Zhou, T. Birker, Generalized shape optimization without homogenization, Structural Optimization 4 (3-4) (1992) 250–252.

[94] K.-U. Bletzinger, K. Maute, Towards generalized shape and topology optimization, Engineering Optimization 29 (1-4) (1997) 201–216.

[95] J. A. Samareh, Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization, AIAA journal 39 (5) (2001) 877–884.

[96] M. Diez, E. F. Campana, F. Stern, Karhunen-loève expansion for assessing stochastic subspaces in geometry optimization and geometric uncertainty quantification, Tech. Rep. 481, IIHR–Hydroscience and Engineering, The University of Iowa (January 2012).

[97] M. Diez, X. Chen, E. Campana, Reliability-based robust design optimization for ships in real ocean environment, in: Proceedings of 12th International Conference on Fast Sea Transportation (FAST2013), Amsterdam, The Netherlands, December 2-5, 2013.

[98] V. Bertram, Chapter 1 - introduction, in: V. Bertram (Ed.), Practical Ship Hydrodynamics (Second Edition), second edition Edition, Butterworth-Heinemann, Oxford, 2012, pp. 1 – 39.

[99] P. Bassanini, U. Bulgarelli, E. F. Campana, F. Lalli, The wave resistance problem in a boundary integral formulation, Surveys on Mathematics for Industry 4 (1994) 151–194.

[100] C. W. Dawson, A practical computer method for solving ship-wave problems, in: Proceedings of the 2nd International Conference on Numerical Ship Hydrodynamics, Berkeley, 1977, pp. 30–38.

[101] J. Telste, A. Reed, Calculation of transom stern flows, in: Proceedings of the Sixth International Conference on Numerical Ship Hydrodynamics, 1994, pp. 78–92.

[102] H. Schlichting, K. Gersten, Boundary-Layer Theory, Springer-Verlag, Berlin, 2000.

[103] W. G. Meyers, A. E. Baitis, SMP84: improvements to capability and prediction accuracy of the standard ship motion program SMP81, Tech. Rep. SPD-0936-04, David Taylor Naval Ship Research and Development Center (September 1985).

[104] J. Huang, P. M. Carrica, F. Stern, Semi-coupled air/water immersed boundary approach for curvilinear dynamic overset grids with application to ship hydrodynamics, International Journal for Numerical Methods in Fluids 58 (6) (2008) 591–624.

[105] A. Levy, A. Montalvo, S. Gomez, A. Galderon, Topics in Global Optimization, Lecture Notes in Mathematics No. 909, Springer-Verlag, 1981.

[106] S. Lucidi, M. Piccioni, Random tunneling by means of acceptance-rejection sampling for global optimization, Journal of optimization theory and applications 62 (2) (1989) 255–277.

[107] E. F. Campana, G. Fasano, A. Pinto, Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization, Journal of Global Optimization 48 (3) (2010) 347–397.

[108] T. T. Wong, W. S. Luk, P. A. Heng, Sampling with hammersley and halton points, Journal of Graphics Tools 2 (2) (1997) 9–24.

[109] A. Carlisle, G. Dozier, An off-the-shelf pso, in: Proceding of the Workshop on Particle Swarm Optimization, 2001, pp. 1–6.

[110] M. Clerc, Stagnation analysis in particle swarm optimization or what happens when nothing happens, Tech. rep. (2006).
URL `http://hal.archives-ouvertes.fr/hal-00122031`

[111] D. Peri, F. Tinti, A multistart gradient-based algorithm with surrogate model for global optimization, Communications in Applied and Industrial Mathematics 3 (1).

[112] M. Clerc, Confinements and biases in particle swarm optimization, Tech. rep. (2006).
URL `http://clerc.maurice.free.fr/pso`

[113] D. Bratton, J. Kennedy, Defining a standard for particle swarm optimization, in: Proc. IEEE Swarm Intell, 2007, pp. 120–127.

[114] S. Helwig, J. Branke, S. Mostaghim, Experimental analysis of bound handling techniques in particle swarm optimization, IEEE Transactions on Evolutionary computation 17 (2) (2013) 9–24.

[115] J. Moré, S. Wild, Benchmarking derivative-free optimization algorithms, SIAM Journal on Optimization 20 (1) (2009) 172–191.

[116] M. Diez, A. Serani, E. F. Campana, O. Goren, K. Sarioz, D. B. Danisman, G. Grigoropoulos, E. Aloniati, M. Visonneau, P. Queutey, F. Stern, Multi-objective hydrodynamic optimization of the DTMB 5415 for resistance and seakeeping, in: Proceedings of the 13th International Conference on Fast Sea Transportation, FAST 2015, Washington, D.C., USA, 2015.

[117] T. Castiglione, F. Stern, S. Bova, M. Kandasamy, Numerical investigation of the seakeeping behavior of a catamaran advancing in regular head waves, Ocean Engineering 38 (16) (2011) 1806 – 1822.

[118] B. Bouscasse, R. Broglia, F. Stern, Experimental investigation of a fast catamaran in head waves, Ocean Engineering 72 (2013) 318 – 330.

[119] R. Broglia, S. Zaghi, E. Campana, M. Visonneau, P. Queutey, T. Dogan, H. Sadat-Hosseini, F. Stern, E. Milanov, CFD validation for DELFT 372 catamaran in static drift conditions, including onset and progression analysis, in: Proceedings of the 5th World Maritime Technology Conference, Rhode Island, USA, November 3-7, 2015.

[120] M. Falchi, M. Felli, S. Grizzi, G. Aloisio, R. Broglia, F. Stern, Spiv measurements around the delft 372 catamaran in steady drift, Experiments in Fluids 55 (11) (2014) 1–20.

[121] T. Castiglione, W. He, F. Stern, S. Bova, Urans simulations of catamaran interference in shallow water, Journal of Marine Science and Technology 19 (1) (2013) 33–51.

[122] W. He, T. Castiglione, M. Kandasamy, F. Stern, Numerical analysis of the interference effects on resistance, sinkage and trim of a fast catamaran, Journal of Marine Science and Technology 20 (2) (2014) 292–308.

[123] R. Broglia, B. Jacob, S. Zaghi, F. Stern, A. Olivieri, Experimental investigation of interference effects for high-speed catamarans, Ocean Engineering 76 (2014) 75 – 85.

[124] H. Sadat-Hosseini, X. Chen, E. Milanov, F. Stern, CFD and system-based prediction of delft catamaran maneuvering in calm water and regular waves, in: Proceedings of 12th International Conference on Fast Sea Transportation (FAST2013), Amsterdam, The Netherlands, December 2-5, 2013.

[125] X. Chen, M. Diez, M. Kandasamy, E. Campana, F. Stern, Design optimization of the waterjet-propelled delft catamaran in calm water using urans, design of experiments, metamodels and swarm intelligence, in: Proceedings of 12th International Conference on Fast Sea Transportation (FAST2013), Amsterdam, The Netherlands, December 2-5, 2013.

[126] F. Stern, J. Longo, R. Penna, A. Olivieri, T. Ratcliffe, H. Coleman, International collaboration on benchmark cfd validation data for surface combatant DTMB model 5415, in: Proceedings of the Twenty-Third Symposium on Naval Hydrodynamics, Val de Reuil, France, September 17-22, 2000.

[127] J. Longo, F. Stern, Uncertainty assessment for towing tank tests with example for surface combatant DTMB model 5415, Journal of Ship Research 49 (1) (2005) 55–68.

[128] Y. Tahara, D. Peri, E. F. Campana, F. Stern, Computational fluid dynamics-based multiobjective optimization of a surface combatant using a global optimization method, Journal of Marine Science and Technology 13 (2008) 95–116.

[129] A. Olivieri, F. Pistani, A. Avanzini, F. Stern, R. Penna, Towing tank, sinkage and trim, boundary layer, wake, and free surface flow around a naval combatant insean 2340 model, Tech. rep., DTIC (2001).

[130] T. Xing, F. Stern, Factors of safety for richardson extrapolation, Journal of Fluids Engineering 132 (6) (2010) 061403.

[131] K. W. H. Eggers, S. D. Sharma, L. W. Ward, An assessment of some experimental methods for the determination of the wavemaking characteristics of a ship form, Tech. Rep. 184, Institut für Schiffbau der Uniuversität Hamburg (1967).

[132] U. Iemma, L. Morino, M. Diez, Digital holography and karhunen–loève decomposition for the modal analysis of two-dimensional vibrating structures, Journal of sound and vibration 291 (1) (2006) 107–131.

[133] U. Iemma, M. Diez, L. Morino, An extended karhunen-loève decomposition for modal identification of inhomogeneous structures, Journal of vibration and acoustics 128 (2006) 357.

[134] M. Irvine Jr., J. Longo, F. Stern, Pitch and heave tests and uncertainty assessment for a surface combatant in regular head waves, Journal of Ship Research 52 (2) (2008) 146–163.

[135] M. Kandasamy, P. Wu, S. Zalek, D. Karr, , S. Bartlett, L. Nguyen, F. Stern, CFD based hydrodynamic optimization and structural analysis of the hybrid ship hull, in: SNAME, 2014.