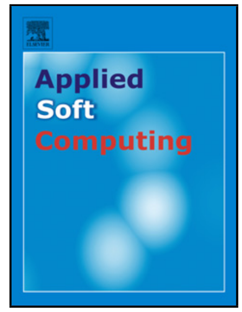
Accepted Manuscript



Title: Formulation and parameter selection of multi-objective deterministic particle swarm for simulation-based optimization

Author: Riccardo Pellegrini Andrea Serani Cecilia Leotardi Umberto Iemma Emilio F. Campana Matteo Diez

PII:	S1568-4946(17)30266-1
DOI:	http://dx.doi.org/doi:10.1016/j.asoc.2017.05.013
Reference:	ASOC 4217

To appear in: *Applied Soft Computing*

Received date:	7-9-2016
Revised date:	30-4-2017
Accepted date:	5-5-2017

Formulation and parameter selection of multi-objective deterministic particle swarm for simulation-based optimization

Riccardo Pellegrini[a,b], Andrea Serani[a], Cecilia Leotardi[a], Umberto Iemma[b], Emilio F. Campana[a], Matteo Diez[a,*]

[a] *CNR-INSEAN, National Research Council–Marine Technology Research Institute, Rome, Italy*
[b] *Department of Mechanical and Industrial Engineering, Roma Tre University, Rome, Italy*

Abstract

Global derivative-free deterministic algorithms are particularly suitable for simulation-based optimization, where often the existence of multiple local optima cannot be excluded a priori, the derivatives of the objective functions are not available, and the evaluation of the objectives is computationally expensive, thus a statistical analysis of the optimization outcomes is not practicable. Among these algorithms, particle swarm optimization (PSO) is advantageous for the ease of implementation and the capability of providing good approximate solutions to the optimization problem at a reasonable computational cost. PSO has been introduced for single-objective problems and several extension to multi-objective optimization are available in the literature. The objective of the present work is the systematic assessment and selection of the most promising formulation and setup parameters of multi-objective deterministic particle swarm optimization (MODPSO) for simulation-based problems. A comparative study of six formulations (varying the definition of cognitive and social attractors) and three setting parameters (number of particles, initialization method, and coefficient set) is performed using 66 analytical test problems. The number of objective functions range from two to three and the number of variables from two to eight, as often encountered in simulation-based engineering problems. The desired Pareto fronts are convex, concave, continuous, and discontinuous. A full-factorial combination of formulations and parameters is investigated, leading to more than 60,000 optimization runs, and assessed by two performance metrics. The most promising MODPSO formulation/parameter is identified and applied to the hull-form optimization of a high-speed catamaran in realistic ocean conditions. Its performance is finally compared with four stochastic algorithms, namely three versions of multi-objective PSO and the genetic algorithm NSGA-II.

Keywords: Multi-objective optimization; derivative-free optimization; global optimization; deterministic particle swarm optimization; simulation-based optimization

1. Introduction

Simulation-based optimization has a peculiar set of features that brings unique challenges to the practicability of the optimization process and the identification of the optimization solutions. The objectives usually derive from complex simulations, which solve partial differential equations, and are computationally very expensive. Therefore, the optimization process needs to rely on a relatively small number of objective evaluations. In most applications, black box tools are used for the simulations and the objective derivatives are not available. Their evaluation through finite differences is often very critical due to the residuals associated to the simulation solutions, which introduce noise in the simulation output. Additionally, the existence of local minima cannot be excluded a priori. For these reasons, global derivative-free algorithms represent an advantageous option for the solution of simulation-based optimization problems.

*Corresponding author. Email: matteo.diez@cnr.it

Preprint submitted to Applied Soft Computing *May 9, 2017*

A large number of global derivative-free algorithms available in the literature are formulated as stochastic optimization methods and make use of random coefficients. In the context of simulation-based applications (with high-fidelity simulation tools), the computational cost of the optimization makes the statistical analysis of the optimization outcomes (required by stochastic algorithms and involving a large number of optimization runs) not practicable, suggesting the adoption of deterministic methods (e.g., [1]). Generally, this consideration holds also in the area of metamodel (or surrogate) based optimization [2, 3], where an approximate representation of the objective function is interposed between the simulation tool and the optimizer. Developing and testing efficient metamodels for optimization require benchmark solutions, provided by the optimzer connected directly to the simulation tool. These solutions are attainable only if efficient and possibly deterministic optimizers are available.

Particle swarm optimization (PSO) was originally introduced by Kennedy and Eberhart [4], based on the social-behavior metaphor of a flock of birds or a swarm of bees searching for food, and belongs to the class of global derivative-free metaheuristic algorithms for single-objective optimization. The algorithm makes use of cognitive and social attractors based on individual and population optima, in order to steer the particle swarm dynamics. For its ease of implementation and capability of providing good approximate solutions to the optimization problem at a reasonable computational cost, PSO has been studied and further developed by a number of authors [5, 6, 7, 8, 9] and successfully applied in engineering optimization [10, 11, 12]. PSO has been extended to multi-objective optimization (MOPSO) by Moore and Chapman [13]. Generally, MOPSO extends the concept of cognitive and social attractors to the multi-objective context, using individual and population non-dominated solution sets (Pareto solutions), sub-swarms, or aggregate objective functions.

Pareto-dominance approaches select cognitive and social attractors from individual and population non-dominated solution sets. Coello et al. [14] randomly select the cognitive attractor among the individual non-dominated solutions. The social attractor is randomly selected from the most isolated regions of the population non-dominated solution set. Fieldsend et al. [15] select the cognitive attractor among the particle non-dominated solutions, whereas the social attractor is selected from an archive structure. Mostaghim and Teich [16] select the cognitive attractor as the particle latest non-dominated solution, whereas the social attractor is defined by the introduction of the Sigma method. Raquel and Naval [17] select the cognitive attractor as the particle oldest non-dominated solution. The social attractor is randomly selected among less-crowded solutions from the population non-dominated solution set. The method makes use of the crowding distance (CD), already used in the non-dominated sorted genetic algorithm (NSGA-II [18]). Nebro et al. [19] proposed a speed-constrained multi-objective PSO (SMPSO). It selects the cognitive attractor randomly getting two points from the population non-dominated solution set and selecting the most isolated between them. The social attractor is selected among less-crowded points from the population non-dominated solution set using a binary tournament. A constriction coefficient is used to limit the particle velocity. Garcia et al. [20] proposed a MOPSO formulation that selects the attractors based on the contribution of the non-dominated solutions to the hypervolume metric [21]. Cognitive/social attractors are randomly selected among the solutions with the least/highest contribution, respectively. Zheng et al. [22] proposed a MOPSO formulation for classification rule mining (MOPSO-CRM). This uses a single attractor for each particle, which is defined depending on whether the particle belongs to the population non-dominated solution set or not. Hu and Yen [23] proposed a MOPSO formulation that selects the cognitive and social attractors based on the solution attributes, evaluated by the parallel cell coordinate system technique introduced in [24]. Adaptive inertia, social, and cognitive coefficients are used.

Sub-swarm approaches make use of sub swarms for exploring the design space, allowing different dynamics for each sub-swarm. Parsopoulos et al. [25] select the cognitive attractor as the optimizer of the objective function explored by the current sub-swarm. The social attractor is the optimizer of a different objective function, explored by another sub-swarm. The algorithm, namely vector evaluated particle swarm optimization (VEPSO), has been inspired by the concept of the vector evaluated genetic algorithm [26]. Peng and Zhang [27] decompose the multi-objective problem into single-objective sub-problems by the Tchebycheff decomposition method. The method selects the cognitive attractor as the optimizer of an aggregated function. The social attractor is selected in the neighborhood of the particle.

Comprehensive surveys on MOPSO variants have been provided in [28] and more recently in [29]. Moreover, effective applications of MOPSO can be found in several engineering fields, such as aerospace [30], civil

[31], electronic [32], industrial [33], and naval [34].

Most PSO formulations (both single- and multi-objective) include stochastic methods and/or random coefficients. This implies that in order to assess the algorithm performace, statistically significant results need to be produced, through extensive numerical campaigns. Such an approach is often too expensive (from the computational viewpoint) and therefore not practicable in simulation-based optimization (especially when computationally expensive solvers are used). For this reason, efficient deterministic approaches, namely deterministic PSO (DPSO) [1, 35] and multi-objective deterministic PSO (MODPSO) [36, 37] have been developed and successfully applied for simulation-based optimization. In deterministic algorithms (both single- and multi-objective), the swarm diversity depends on the swarm dynamics provided by the combination of formulation and parameters. During the swarm evolution each particle is attracted by diverse positions, based on the cognitive and social experience iteration by iteration. In most problems, this is generally sufficient to maintain the swarm dynamics and provide reasonable solutions. Chen et al. [38] discussed the effectiveness of DPSO, comparing to random PSO for a hull-form optimization problem. Recently, Serani and Diez [39] proposed a statistical analysis of random PSO for a set of 100 problems, with comparison to DPSO.

A systematic study for the parameter selection of the single-objective DPSO has been presented by Serani et al. [35], for box-constrained simulation-based problems in ship hydrodynamics. The most promising selection of number of particles, initialization approach, coefficient set, and box constraint method has been investigated and discussed. Extending the study to MODPSO needs to assess and discuss: (a) the algorithm formulation for multi-objective problems and, similarly to the single-objective DPSO, (b) the number of particles interacting during the optimization, (c) the initialization of the particles in terms of initial location and velocity, and (d) the set of coefficients defining the cognitive and social behavior of the swarm dynamics. A preliminary systematic study of MODPSO has been presented in [40]. Nevertheless, discussions and applications of MODPSO in simulation-based problems is still limited, lacking a systematic and a comparative analysis of (a), (b), (c), and (d).

The objective of the present work is a systematic study of the MODPSO performance conditional to the algorithm formulation of the cognitive/social attractors and the parameter selection, with focus on simulation-based applications. Four deterministic variants of multi-objective swarm methods are introduced here and included in the analysis, where a total of six formulations are assessed and compared. The present study is an extension to multi-objective problems of the work presented by Serani et al. [35].

The analysis approach includes a preliminary systematic study on 66 analytical test problems from the literature [41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51], with a number of variables ranging from two to eight and a number of objectives from two to three. The problems are characterized by convex and non-convex, continuous and discontinuous Pareto fronts. A full-factorial combination is investigated of: (a) algorithm formulation, (b) number of particles, (c) particles initialization, and (d) coefficient set. Six MODPSO formulations are addressed, where:

1. the cognitive attractor is the closest point to the particle of the individual Pareto solutions; the social attractor is the closest point to the particle of the population Pareto solutions; distances are evaluated in the variable space (domain)

2. the cognitive attractor is the closest point to the particle of the individual Pareto solutions; the social attractor is the closest point to the particle of the population Pareto solutions; distances are evaluated in the objective space (codomain)

3. the cognitive attractor is the individual best position, based on an aggregate objective function; the social attractor is the closest point to the particle of the population Pareto solutions; the distance is evaluated in the variable space (domain)

4. the cognitive attractor is the individual best position, based on an aggregate objective function; the social attractor is the closest point to the particle of the population Pareto solutions; the distance is evaluated in the objective space (codomain)

5. the cognitive attractor is selected as the particle oldest-found non-dominated solution; the social attractor is the closest point to the particle of a subset of the population Pareto solutions with largest crowding distance; the distance is evaluated in the variable space (domain)

6. sub-population deterministic formulation following the VEPSO approach

The first formulation (MODPSO1) has been presented in [37] and applied to the robust optimization of a bulk carrier. This is extended herein by embedding the formulation in the codomain (MODPSO2). The third formulation (MODPSO3) is taken from [52] and extended to the codomain in MODPSO4. The fifth formulation (MODPSO5) is a current deterministic implementation of the original MOPSO-CD, presented in [17] and considered by García et al. [20] a state-of-the-art algorithm for Pareto-dominance approaches. Finally, MODPSO6 is a current deterministic implementation of the original VEPSO formulation, presented in [25] and used here as a representative of sub-swarm methods.

The parameter selection includes the number of particles, parametrized as proportional to the number of design variables and objective functions. The initialization of the particle position and velocity is defined by a deterministic distribution (Hammersley sequence sampling, HSS, [53]). The coefficient sets are chosen from literature [6, 7, 52, 9, 37]. The box constraints are handled by a semi-elastic wall type approach [35]. The full-factorial combination of MODPSO formulation, parameters, and analytical test problems results in more than 60,000 optimization runs, performed on an Intel Xeon E5-1620 v2 3.70GHz.

The algorithm performances are assessed by three general criteria, based on the number of non-dominated solutions (capacity), the evolution of the Pareto solution (convergence), and the variety of the solution (diversity). These criteria are quantitatively evaluated by the following three metrics [54]:

i) ratio of reference points, $\mathrm{C1_R}$ [55] (capacity),

ii) hypervolume, HV [21] (convergence and diversity),

iii) combination of $\mathrm{C1_R}$ and HV (capacity, convergence, and diversity).

Based on these metrics, the most promising MODPSO formulation and parameter setup are identified and applied to a hull-form optimization of a high-speed catamaran in realistic ocean environment, sailing in head waves in the North Pacific Ocean, including stochastic sea state and speed [34]. The problem is formulated as a multi-objective optimization aimed at (1) the reduction of the expected value of the mean total resistance in irregular waves, at variable speed and (2) the increase of the ship operability, with respect to a set of motion-related constraints. The design space is a four-dimensional representation of shape modifications, based on the Karhunen-Loève expansion (KLE) of free-form deformations of the original hull [56]. For the sake of the current study, the optimization is performed on a metamodel, based on stochastic radial basis function [57] and trained by an unsteady Reynolds averaged Navier-Stokes (URANS) equations solver. Finally, the results are compared with a stochastic version of the most promising MODPSO, MOPSO-CRM, MOPSO-CD, and NSGA-II.

The paper is organized as follows. Section 2 presents the multi-objective problem formulation and associated definitions. Section 3 briefly recalls the single-objective particle swarm optimization algorithm, whereas Section 4 presents the multi-objective extensions of the formulation. Section 5 describes the setting parameters used for MODPSO. The performance metrics are presented in Section 6, whereas the optimization problems are described in Section 7. The discussion of the results and the final conclusions are included in Sections 8 and 9, respectively.

2. Optimization problem formulation

The multi-objective minimization problem is formulated as

$$\text{minimize } \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\}^T, \quad m = 1, ..., N_{\rm of}, \quad \mathbf{x} \in \mathbb{R}^{N_{\rm dv}} \tag{1}$$

where N_{of} is the number of objective functions $f_m(\mathbf{x})$, and $\mathbf{x}$ is the vector collecting the N_{dv} variables. Geometric and/or functional constraints, if required, may be applied and included in the problem of Eq. 1 as

$$\begin{aligned} z_i(\mathbf{x}) &\leq 0, \quad \text{with} \quad i = 1, \ldots, I \\ h_j(\mathbf{x}) &= 0, \quad \text{with} \quad j = 1, \ldots, J \end{aligned} \tag{2}$$

where $z_i(\mathbf{x})$ are the inequality constraints and $h_j(\mathbf{x})$ are the equality constraints, defining the feasible solution set as

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^{N_{\text{dv}}} \ | \ [\cap_i^I z_i(\mathbf{x}) \leq 0] \wedge [\cap_j^J h_j(\mathbf{x}) = 0]\} \tag{3}$$

The solution of Eq. 1 is the locus of non-dominated feasible solutions, represented in the variable domain by the Pareto solution set

$$\mathcal{PS} = \{\mathbf{x} \in \mathcal{X} \ | \ \mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{y}), \forall \mathbf{y} \in \mathcal{X}\} \tag{4}$$

In the objective function space, the locus is represented by the Pareto front

$$\mathcal{PF} = \{\mathbf{f}(\mathbf{x}) \ : \ \mathbf{x} \in \mathcal{PS}\} \tag{5}$$

In the following, the approximate solution set (set of non-dominated solutions represented either in the variable or function space) achieved by the optimizer at a specific iteration is indicated by $\mathcal{S}$

$$\mathcal{S} = \{(\mathbf{x}, \mathbf{s}) \ : \ \mathbf{s} = \mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{y}), \forall \mathbf{y} \in \mathcal{X}\} \tag{6}$$

Similarly, the approximate reference solution used for the performance evaluation and assessed by numerical experiments is indicated by $\mathcal{R}$.

3. Particle swarm optimization

PSO algorithm was originally introduced in [4] and is based on the social-behaviour metaphor of a flock of birds or a swarm of bees searching for food. PSO belongs to the class of metaheuristic algorithms for single-objective derivative-free global optimization. PSO has been formulated in [5] as

$$\begin{cases} \mathbf{v}_i^{k+1} = \chi \left[\mathbf{v}_i^k + c_1 r_{1,i}^k \left(\mathbf{p}_i - \mathbf{x}_i^k\right) + c_2 r_{2,i}^k \left(\mathbf{g} - \mathbf{x}_i^k\right)\right] \\ \mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{cases} \tag{7}$$

where $\mathbf{v}_i^k$ and $\mathbf{x}_i^k$ are the velocity and the position, respectively, of the i-th particle at the k-th iteration. The parameter χ, namely the constriction factor, has been introduced in [5] to improve the convergence of PSO; c_1 and c_2 are the cognitive and social learning rate, respectively; $r_{1,i}^k$ and $r_{2,i}^k$ are two random numbers in $[0, 1]$. Finally $\mathbf{p}_i$ is the personal best position ever visited by the i-th particle and $\mathbf{g}$ the global best position ever visited among all particles. PSO formulation in Eq. 7 makes use of random coefficients, in order to enhance the variety of the swarm dynamics. This property implies that statistically significant results can be obtained only through extensive numerical campaigns. Such an approach might be too expensive in simulation-based optimization for real industrial applications, therefore a deterministic formulation of PSO (DPSO) has been introduced in [1]. Setting $r_{1,i}^k = r_{2,i}^k = 1$ yields

$$\begin{cases} \mathbf{v}_i^{k+1} = \chi \left[\mathbf{v}_i^k + c_1 \left(\mathbf{p}_i - \mathbf{x}_i^k\right) + c_2 \left(\mathbf{g} - \mathbf{x}_i^k\right)\right] \\ \mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{cases} \tag{8}$$

4. Multi-objective deterministic PSO (MODPSO)

The single-objective DPSO algorithm is extended to multi-objective problems [36] as

$$\begin{cases} \mathbf{v}_i^{k+1} = \chi \left[\mathbf{v}_i^k + c_1 \left(\mathbf{p}_i - \mathbf{x}_i^k \right) + c_2 \left(\mathbf{g}_i - \mathbf{x}_i^k \right) \right] \\ \mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{cases} \tag{9}$$

where the global (social) best position for the single-objective DPSO $\mathbf{g}$ is redefined as $\mathbf{g}_i$. The subscript "i" implies that each particle has its own social attractor.

It is possible to prove that the necessary (but possibly not sufficient) conditions which ensure that the trajectory of each particle does not diverge [58], are

$$\begin{cases} 0 < |\chi| < 1 \\ 0 < \omega < 2(\chi + 1) \end{cases} \tag{10}$$

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

$$\beta = \frac{\omega}{2(\chi + 1)} \tag{11}$$

and assuming $\chi > 0$ as usually in literature, the conditions of Eq. 10 reduce to

$$\begin{cases} 0 < \chi < 1 \\ 0 < \beta < 1 \end{cases} \tag{12}$$

In the following, six formulations of MODPSO are presented and compared. Four of them have been selected from literature, whereas two variants are proposed herein. The formulations differ for the approach used to define the cognitive and social attractors ($\mathbf{p}_i$ and $\mathbf{g}_i$), as presented in the following.

4.1. Pareto solution formulations

Diez and Peri [37] have presented a Pareto solution formulation, where the cognitive and social attractors are defined as follows:

- $\mathbf{p}_i$ is the closest point to the i-th particle of the personal solution set $\mathcal{S}_{\mathrm{p},i}$;
- $\mathbf{g}_i$ is the closest point to the i-th particle of the solution set $\mathcal{S}$;

where $\mathcal{S}_{\mathrm{p},i}$ is the set of all non-dominated solutions ever visited by the i-th particle. This formulation is referred to as MODPSO1 and presented in Alg. 1.

Algorithm 1 MODPSO1

Initialize a swarm of N_p particles
while ($k <$ Max number of iterations) **do**
  for $i = 1, N_\mathrm{p}$ **do**
    Evaluate $\mathbf{f}(\mathbf{x}_i^k)$
    Evaluate $\mathcal{S}_{\mathrm{p},i}$
  end for
  Evaluate $\mathcal{S}$
  for $i = 1, N_\mathrm{p}$ **do**
    Evaluate cognitive attractor, $\mathbf{p}_i = \mathrm{argmin}||\mathbf{x}_i^k - \mathbf{x}||, \mathbf{x} \in \mathcal{S}_{\mathrm{p},i}$
    Evaluate social attractor, $\mathbf{g}_i = \mathrm{argmin}||\mathbf{x}_i^k - \mathbf{x}||, \mathbf{x} \in \mathcal{S}$
  end for
  Update particle velocities $\mathbf{v}_i^{k+1}$
  Update particle positions $\mathbf{x}_i^{k+1}$
end while
Output solution set

A further variant of MODPSO1 is presented in this work, namely MODPSO2. Differently from MODPSO1, the distance of the i-th particle to the solution set is evaluated in the objective function space (see Alg. 2).

Algorithm 2 MODPSO2

Initialize a swarm of N_p particles
while ($k <$ Max number of iterations) **do**
  for $i = 1, N_\mathrm{p}$ **do**
    Evaluate $\mathbf{f}(\mathbf{x}_i^k)$
    Evaluate $\mathcal{S}_{\mathrm{p},i}$
  end for
  Evaluate $\mathcal{S}$
  for $i = 1, N_\mathrm{p}$ **do**
    Evaluate cognitive attractor, $\mathbf{p}_i = \mathrm{argmin}||\mathbf{f}(\mathbf{x}_i^k) - \mathbf{f}(\mathbf{x})||, \mathbf{x} \in \mathcal{S}_{\mathrm{p},i}$
    Evaluate social attractor, $\mathbf{g}_i = \mathrm{argmin}||\mathbf{f}(\mathbf{x}_i^k) - \mathbf{f}(\mathbf{x})||, \mathbf{x} \in \mathcal{S}$
  end for
  Update particle velocities $\mathbf{v}_i^{k+1}$
  Update particle positions $\mathbf{x}_i^{k+1}$
end while
Output solution set

4.2. Combined Pareto solution/aggregate objective formulations

Campana and Pinto [52] have presented a formulation variant, combining the Pareto solution with an aggregate objective function. The cognitive and social attractors are defined as follows:

- $\mathbf{p}_i$ is the personal minimizer of the aggregated function $F(\mathbf{x}_i)$;
- $\mathbf{g}_i$ is the closest point to the i-th particle of the solution set $\mathcal{S}$.

The aggregated function $F(\mathbf{x}_i)$ is defined as:

$$F(\mathbf{x}_i) = \sum_{m=1}^{N_\mathrm{of}} w_m f_m(\mathbf{x}_i) \tag{13}$$

where w_m is the weight associated to the m-th objective function. Herein, the objective function weights w_m are set equal to one and the distance of the i-th particle to the solution set is computed in the variable space. This implementation is referred to as MODPSO3 (see Alg. 3).

Algorithm 3 MODPSO3

Initialize a swarm of N_p particles
while ($k <$ Max number of iterations) **do**
  for $i = 1, N_\mathrm{p}$ **do**
    Evaluate $\mathbf{f}(\mathbf{x}_i^k)$
    Evaluate $\mathcal{S}_{\mathrm{p},i}$
  end for
  Evaluate $\mathcal{S}$
  for $i = 1, N_\mathrm{p}$ **do**
    Evaluate cognitive attractor, $\mathbf{p}_i = \mathrm{argmin}[F(\mathbf{x}_i)]$
    Evaluate social attractor, $\mathbf{g}_i = \mathrm{argmin}||\mathbf{x}_i^k - \mathbf{x}||, \mathbf{x} \in \mathcal{S}$
  end for
  Update particle velocities $\mathbf{v}_i^{k+1}$
  Update particle positions $\mathbf{x}_i^{k+1}$
end while
Output solution set

A further variant of MODPSO3 is presented in this work, namely MODPSO4. Differently from MODPSO3, the distance of the i-th particle to the solution set is computed in the objective function space (see Alg. 4).

Algorithm 4 MODPSO4

```
Initialize a swarm of N_p particles
while (k < Max number of iterations) do
    for i = 1, N_p do
        Evaluate f(x_i^k)
        Evaluate S_{p,i}
    end for
    Evaluate S_g
    for i = 1, N_p do
        Evaluate cognitive attractor, p_i = argmin[F(x_i)]
        Evaluate social attractor, g_i = argmin||f(x_i^k) − f(x)||, x ∈ S
    end for
    Update particle velocities v_i^{k+1}
    Update particle positions x_i^{k+1}
end while
Output solution set
```

4.3. Combined Pareto solution/crowding distance formulation

Raquel e Naval [17] have presented a variant of MOPSO, which enhances the diversity of the swarm including the crowding distance as a criterion to select the social attractor. A deterministic version of the algorithm is herein used, obtained eliminating the random coefficients of the attractors, the mutation of the particles, and introducing a deterministic criterion for the initialization and the selection of the social attractor. The cognitive and social attractors are defined as follows:

- $\mathbf{p}_i$ is the best position of the i-th particle;
- $\mathbf{g}_i$ is the closest point to the i-th particle of the best 10% of solution set $\mathcal{S}$, ordered in descending crowding distance values of the points of $\mathcal{S}$.

This implementation is referred to as MODPSO5 (see Alg. 5).

Algorithm 5 MODPSO5

```
Initialize a swarm of N_p particles
Initialize cognitive attractor, p_i = x_i^0
while (k < Max number of iterations) do
    for i = 1, N_p do
        Evaluate f(x_i^k)
        Evaluate S_{p,i}
    end for
    Evaluate S
    Order S in descending crowding distance values
    According to the crowding distance values, select the best 10% of S (S_10)
    for i = 1, N_p do
        Evaluate cognitive attractor p_i
        if x_i^k dominate p_i then
            p_i = x_i^k
        end if
        Evaluate social attractor, g_i = argmin||x_i^k − x||, x ∈ S_10
    end for
    Update particle velocities v_i^{k+1}
    Update particle positions x_i^{k+1}
end while
Output solution set
```

It should be noted that the maximum value of the crowding distance is always assigned to the solution set extrema in order to include them among the feasible attractors.

4.4. Sub-population formulation

Parsopulos et al. [25] have presented a MOPSO variant based on a sub-population formulation, known as vector evaluated PSO (VEPSO). The particle swarm is divided into $N_{\rm of}$ sub-swarm. Here, a deterministic variant of the formulation is used (namely MODPSO6), with sub-swarms exchanging information in a ring connection (see Fig. 1 and Alg. 6 from line 10 to 17). Cognitive and social attractors of the m-th sub-swarm are defined as follows:

- $\mathbf{p}_{i,m}$ is the personal minimizer of the objective function $f_m(\mathbf{x}_{i,m})$,
- $\mathbf{g}_{i,m}$ is the global minimizer of the objective function $f_{m-1}(\mathbf{x}_{i,m-1})$, provided by the $(m-1)$-th sub swarm.

The method is presented in Alg. 6.

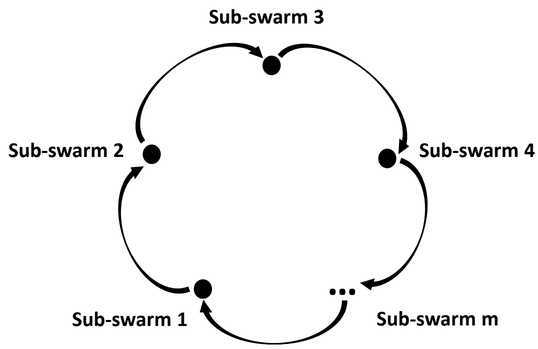


Figure 1: Ring connection scheme for MODPSO6

Algorithm 6 MODPSO6

Initialize $M = N_{\mathrm{of}}$ sub-swarms
while ($k <$ Max number of iterations) **do**
  for $m = 1, M$ **do**
    for $i = 1, N_{\mathrm{p}}/M$ **do**
      Evaluate $f_m(\mathbf{x}_{i,m}^k)$
      Evaluate cognitive attractor, $\mathbf{p}_{i,m} = \mathrm{argmin}[f_m(\mathbf{x}_{i,m})]$
    end for
    Evaluate auxiliary social attractor, $\mathbf{a}_m = \mathrm{argmin}[f_m(\mathbf{p}_{i,m})], i = 1, ..., N_{\mathrm{p}}/M$
  end for
  for $i = 1, N_{\mathrm{p}}/M$ **do**
    Define social attractor, $\mathbf{g}_{i,1} = \mathbf{a}_M$
  end for
  for $m = 2, M$ **do**
    for $i = 1, N_{\mathrm{p}}/M$ **do**
      Define social attractor, $\mathbf{g}_{i,m} = \mathbf{a}_{m-1}$
    end for
  end for
  for $m = 1, M$ **do**
    Update particle velocities $\mathbf{v}_{i,m}^{k+1}$
    Update particle positions $\mathbf{x}_{i,m}^{k+1}$
  end for
  Evaluate $\mathcal{S}$
end while
Output solution set

5. MODPSO parameters and setup

The MODPSO parameters used in the current analysis are described and discussed in the following subsections. Their full-factorial combination is considered, resulting in a total of 180 MODPSO setups for each algorithm formulation.

5.1. Number of particles

The number of particles (N_p) is selected as

$$N_\mathrm{p} = 2^n N_\mathrm{of} N_\mathrm{dv}, \quad \text{with} \quad n \in \mathbb{N}[1,6] \tag{14}$$

therefore ranging from $2N_\mathrm{of}N_\mathrm{dv}$ to $64N_\mathrm{of}N_\mathrm{dv}$.

5.2. Particles initialization

The initialization of particles location and velocity is set using a deterministic and homogeneous distribution, following HSS [53]. Specifically, three different sub-domains are investigated [35]: (a) the variable domain, (b) the boundary domain, (c) the variable domain and the boundary domain using particles in even amount. Both null and non-null [38] initial velocities are considered for the particles, resulting in six different initializations. Table 1 summarizes the initialization used in this work.

Table 1: Swarm initialization

HSS, over	$\mathbf{v} = 0$	$\mathbf{v} \neq 0$
Domain	A.0	A.1
Domain boundaries	B.0	B.1
Domain and boundaries	C.0	C.1

5.3. Coefficient set

Five coefficient sets, summarized in Tab. 2, are selected from literature and used for the current analysis. The associated values of β are included, and they all satisfy Eq. 12.

Table 2: Coefficient sets

Set ID	Reference	χ	c_1	c_2	β
1	Shi and Eberhart [6]	0.729	2.050	2.050	0.864
2	Trelea [7]	0.600	1.700	1.700	0.638
3	Campana and Pinto [52]	1.000	0.400	1.300	0.425
4	Clerc [9]	0.721	1.655	1.655	0.694
5	Diez and Peri [37]	0.990	0.330	0.660	0.246

5.4. Box constraints

A semi-elastic wall-type approach [35] is used to keep the particles inside the feasible domain. In case the i-th 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), whereas the associated j-th velocity component is defined as follows

$$v_{i,j}^{k+1} = -\frac{v_{i,j}^{k+1}}{\chi\,(c_1 + c_2)} \tag{15}$$

5.5. Number of problem evaluations

One problem evaluation involves one evaluation of each objective function. This parameter directly affects the number of iterations available to perform the optimization. The number of problem evaluation (N_peval) is defined as

$$N_\mathrm{peval} = \nu N_\mathrm{of} N_\mathrm{dv}, \quad \text{where} \quad \nu = 125 \cdot 2^c \quad \text{with} \quad c \in \mathbb{N}[0,4] \tag{16}$$

therefore N_peval ranges between $125N_\mathrm{of}N_\mathrm{dv}$ and $2000N_\mathrm{of}N_\mathrm{dv}$. Consequently, the number of MODPSO iterations, from Eq. 14, is set as

$$N_\mathrm{iter} = \frac{N_\mathrm{peval}}{N_\mathrm{p}} = \frac{125 \cdot 2^c N_\mathrm{dv} N_\mathrm{of}}{2^n N_\mathrm{dv} N_\mathrm{of}} = 125 \cdot 2^{c-n} \tag{17}$$

6. Performance metrics

In the present work, the performance metrics are defined following the classification presented in [54]:

- *capacity* metrics are used to tally the number of non-dominated solution $\mathcal{S}$ that satisfied given predefined requirements;
- *converge-diversity* metrics are used to include convergence and diversity information on a single scale.

Herein, the reference solution set $\mathcal{R}$, for a problem $p \in \mathcal{P}$, is defined as

$$\mathcal{R} = \{(\mathbf{x}, \mathbf{r}) \in \cup_{i=1}^{N_s} \mathcal{S}_i \quad : \quad \mathbf{r} = \mathbf{f}(\mathbf{x}) \prec \mathbf{s}, \forall \mathbf{s}\} \tag{18}$$

where N_s is the number of algorithm formulations/setups.

6.1. Capacity metric

Herein, the *Ratio of Reference Point Found* ($\mathrm{C1_R}$, [55]) is used as capacity metric. It is evaluated by

$$\mathrm{C1_R} = \frac{|\mathcal{S} \cap \mathcal{R}|}{|\mathcal{R}|} \tag{19}$$

This metric quantifies the contribution of a solution set $\mathcal{S}$ to the reference solution set $\mathcal{R}$ (see Fig. 2a). High values of $\mathrm{C1_R}$ correspond to better performance.

6.2. Convergence-diversity metrics

The widely-used *HyperVolume* (HV), introduced in [21], is defined as

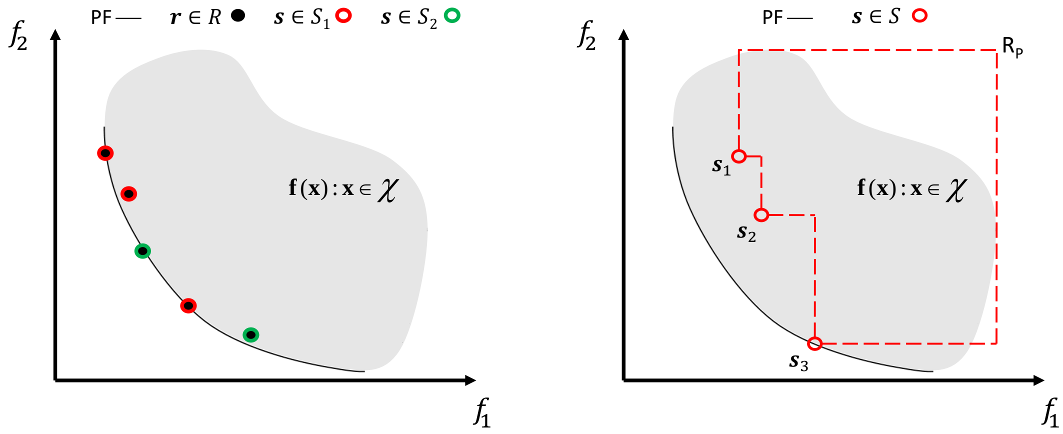
$$\mathrm{HV}(\mathcal{S}, \mathcal{R}) = \mathrm{volume}\left(\bigcup_{i=1}^{|\mathcal{S}|} v_i\right) \tag{20}$$

This metric gives the hypervolume (in the codomain) dominated by the solution set $\mathcal{S}$. An example is given in Fig. 2b, where the HV is the area bounded by $\mathrm{S_1S_2S_3R_pS_1}$. $\mathrm{R_p}$ is the anti-ideal point of $\mathcal{R}$ [54].

A *Normalized Hypervolume* (NHV) is introduced herein as

$$\mathrm{NHV} = \frac{\mathrm{HV}(\mathcal{S}, \mathcal{R})}{\mathrm{HV}(\mathcal{R}, \mathcal{R})} \tag{21}$$

where $\mathrm{HV}(\mathcal{R}, \mathcal{R})$ is the hypervolume of the reference solution set. High values of NHV correspond to better performance.



(a) Ratio of Reference Point Found, $C1_{\mathrm{R}}$

(b) Hypervolume, HV

Figure 2: Graphical interpretation of $C1_{\mathrm{R}}$ and NHV

6.3. Relative variability

The relative variability $\sigma^2_{r,k}$ [59] is used to assess the impact of formulation and parameters on each performance metric. Defining the algorithm formulation/parameter vector as $\mathbf{t} = \{t_1, \ldots, t_4\}^T \in \mathcal{T}$ (collecting respectively formulation, number of particles, initialization, and coefficient set), the relative performance variability associated to its k-th component is

$$\sigma^2_{r,k} = \frac{\sigma^2_k}{\sum_{k=1}^{|\mathcal{T}|} \sigma^2_k} \tag{22}$$

where

$$\sigma^2_k = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} [\hat{\mu}_k(\omega)]^2 - \left[\frac{1}{|\Omega|} \sum_{\omega \in \Omega} \hat{\mu}_k(\omega) \right]^2 \tag{23}$$

with Ω containing the positions ω assumed by the parameter t_k

$$\hat{\mu}_k(\omega) = \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \bar{\mu}(\mathbf{t}), \qquad \mathcal{B} = \{\mathbf{t} : t_k = \omega\} \tag{24}$$

and

$$\bar{\mu}(\mathbf{t}) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} [\mu(\mathbf{t})]_p \tag{25}$$

where $[\mu(\mathbf{t})]_p$ is the value of any metric ($C1_{\mathrm{R}}$ or NHV) given by the formulation/parameters $\mathbf{t}$, for the problem p.

6.4. Performance profiles

The performance profile is the cumulative distribution function for a specific performance metric μ and is defined as [60]

$$\rho_{\mathbf{t}}(\tau) = \frac{1}{|\mathcal{P}|} \mathrm{size}\, \{p \in \mathcal{P} : [r(\mathbf{t})]_p \leq \tau\} \tag{26}$$

where $[r(\mathbf{t})]_p$ is the performance ratio

$$[r(\mathbf{t})]_p = \frac{[\mu(\mathbf{t})]_p}{\min\{[\mu(\mathbf{t})]_p : \mathbf{t} \in \mathcal{T}\}} \tag{27}$$

and $\tau \geq 1$.

The metrics summarized in Tab. 5 are used to compute the performance ratio and the performance profile, namely $C1_{\mathrm{R}}$ and NHV.

It is worth noting that $\rho_{\mathbf{t}}(1)$ gives the number of problems that the formulation/parameters $\mathbf{t}$ solves better than others [60]. Formulation/parameters with a high probability of success within a certain tolerance are identified by high values of $\rho_{\mathbf{t}}(\tau)$ with $\tau > 1$.

6.5. Data profiles

The data profile is a cumulative distribution function for a specific performance metric μ [61, 62], defined as

$$d_{\mathbf{t}}(\nu) = \frac{1}{|\mathcal{P}|}\mathrm{size}\left\{p \in \mathcal{P} : [q(\mathbf{t})]_p \leq \nu\right\} \tag{28}$$

where $[q(\mathbf{t})]_p$ is the number of problem evaluations required for formulation/parameters $\mathbf{t}$ to solve problem p, with a certain accuracy (tolerance) ε.

7. Optimization problems

7.1. Analytical test problems

Sixty six analytical test problems (see Tab. 3), selected from literature [41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51] in order to provide a heterogeneous collection of problems with convex and non-convex, continuous and discontinuous Parent fronts have been used to determine the best performing MODPSO formulation and setup. Test problems with $N_{\mathrm{of}} = 2$ and 3 have been considered, with N_{dv} ranging from 2 to 8. The corresponding occurrences are summarized in Tab. 4.

Table 3: Analytical test problems

p	Name	N_{dv}	N_{of}	Front	Reference	p	Name	N_{dv}	N_{of}	Front	Reference
1	ex005	2	2	Cont convex/concave	[41]	34	LE1	2	2	Cont concave	
2	Kursawe	3	2	Discont convex/concave	[42]	35	LRS1	2	2	Cont convex	
3	Fonseca	2	2	Cont concave	[43]	36	MHHM2	2	3	Cont convex	
4	CL1	4	2	Cont convex	[44]	37	MLF2	2	2	Discont concave	
5	Deb41	2	2	Cont convex		38	MOP2	4	2	Cont concave	
6	Deb512a	2	2	Cont convex		39	MOP3	2	2	Discont convex	
7	Deb512b	2	2	Cont concave		40	MOP4	3	2	Discont convex/concave	
8	Deb512c	2	2	Cont convex	[45]	41	MOP5	2	3	Cont convex/concave	
9	Deb513	2	2	Discont convex/concave		42	MOP6	2	2	Discont convex/concave	
10	Deb521a	2	2	Cont concave		43	MOP7	2	3	Cont convex/concave	
11	Deb521b	2	2	Cont concave		44	SK2	4	2	Discont convex/concave	
12	Jin1	2	2	Cont convex		45	SP1	2	2	Cont convex	
13	Jin2	2	2	Cont convex	[46]	46	SSFYY1	2	2	Cont convex	[50]
14	Jin3	2	2	Cont concave		47	TKLY1	4	2	Cont convex	
15	Jin4	2	2	Cont convex/concave		48	VFM1	2	3	Cont convex	
16	DTLZ1	7	3	Cont convex		49	VU1	2	2	Cont convex	
17	DTLZ1n2	2	2	Cont convex		50	VU2	2	2	Cont convex	
18	DTLZ2n2	2	2	Cont concave		51	WFG2	8	3	Cont concave	
19	DTLZ3n2	2	2	Cont convex	[47]	52	WFG3	8	3	Cont concave	
20	DTLZ4n2	2	2	Cont concave		53	WFG3$_{bis}$	8	3	Cont convex	
21	DTLZ5n2	2	2	Cont concave		54	WFG4	8	3	Cont concave	
22	DTLZ6n2	2	2	Discont convex/concave		55	WFG5	8	3	Cont concave	
23	OKA1	2	2	Cont convex	[48]	56	WFG6	8	3	Cont concave	
24	OKA2	3	2	Cont concave		57	WFG7	8	3	Cont concave	
25	BK1	2	2	Cont convex		58	WFG8	8	3	Cont concave	
26	Far1	2	2	Cont convex/concave		59	WFG9	8	3	Cont concave	
27	I1	8	3	Cont concave		60	lovison1	2	2	Cont convex	
28	I2	8	3	Cont concave		61	lovison2	2	2	Cont concave	
29	I3	8	3	Cont concave	[50]	62	lovison3	2	2	Cont convex	[51]
30	I4	8	3	Cont concave		63	lovison4	2	2	Cont convex	
31	I5	8	3	Discont concave		64	lovison5	3	3	Discont convex	
32	IKK1	2	3	Cont convex		65	lovison6	3	3	Cont convex	
33	IM1	2	2	Cont concave		66	mop_vicente	2	2	Cont convex	-

Table 4: Occurrence of number of variables N_{dv} and objective functions N_{of}

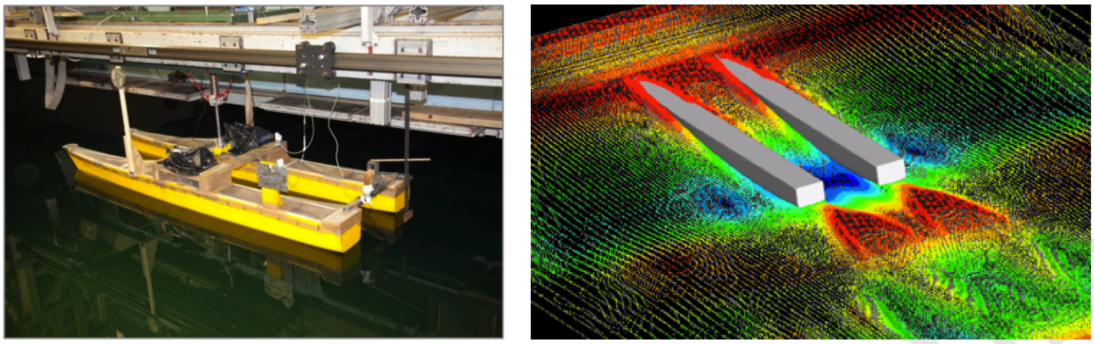
	N_{of}		N_{dv}				
Value	2	3	2	3	4	7	8
Occurrence	44	22	42	5	4	1	14

7.2. Catamaran hull-form optimization

The industrial application presented pertains to the reliability-based robust optimization of the hull form of a 100 m high-speed catamaran [34], sailing in head waves in the North Pacific Ocean. Figure 3 shows the model used at CNR-INSEAN for the experiments and an example of the wave pattern obtained by URANS simulation [63].

The problem is formulated as

$$\begin{aligned} \text{minimize} \quad & \{\varphi_1(\mathbf{x}),\, -\varphi_2(\mathbf{x})\}^T \\ \text{subject to} \quad & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \\ \text{and to} \quad & \varphi_1 \leq 0;\ \ \varphi_2 \geq 0 \end{aligned} \tag{29}$$



(a) (b)

Figure 3: High-speed catamaran: (a) CNR-INSEAN model and (b) URANS wave pattern [63]

where ϕ_1 and ϕ_2 are the expected value of the mean total resistance and the ship operability evaluated in irregular wave for variable sea state and speed, respectively defined as

$$\varphi_1(\mathbf{x}) = \iint_{S,U} \bar{R}_T(\mathbf{x}, S, U)\, p(S,U)\, dU dS \tag{30}$$

$$\varphi_2(\mathbf{x}) = \iint_{S,U} \bigcap_{j=1}^{J} [h_j(\mathbf{x}, S) \leq 0] p(S,U)\, dU dS \tag{31}$$

where $\bar{R}_T$ is the mean value of the total resistance in irregular waves, $\mathbf{x}$ is the design variable vector ($N_{\rm dv} = 4$), S is the sea state, U is the speed, h_j are the motion constraints, and p is the joint probability density function of S and U.

The design optimization problem is taken from [34], and solved by means of stochastic radial-basis functions interpolation (details may be found in [57]) of high-fidelity URANS simulations. Four design variables control global shape modifications of the catamaran hull, based on the Karhunen-Loève expansion of the shape modification vector [56]. The inequalities in Eq. 29 are handled by a linear penalty function method.

8. Numerical results

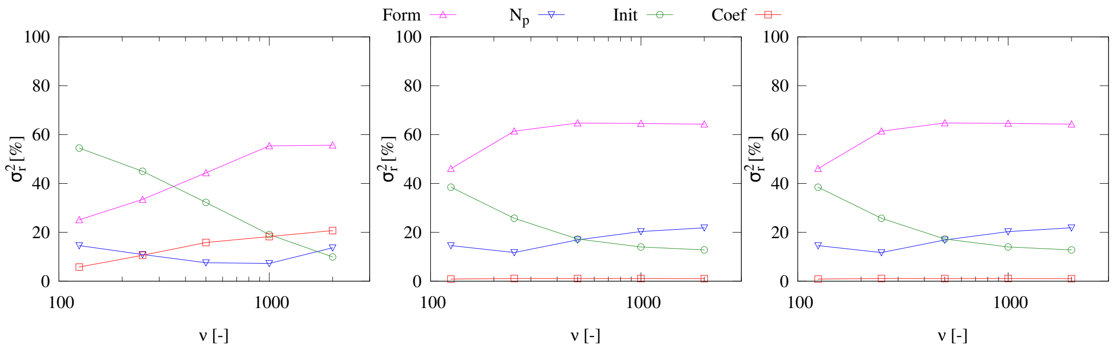
Analytical test problems results are used to define the most promising MODPSO formulation/parameter setup overall, used later for the catamaran optimization. The selection of the best performing formulation/parameter setup is based on the sum of $\mathrm{C1_R}$ and NHV. In order to provide a proper comparison between different problems, with different codomain size, each solution set $\mathcal{S}$ is normalized with respect to the function range, therefore $s_i \in [0,1]$ and the reference point for the computation of HV is $\mathrm{R_P} = \{1\}$. The computation of HV is performed with the code provided by [64]. Table 5 summarizes the criteria used for the analysis of the results. The accuracy used for the computation of the data profiles is also provided.

Table 5: Metrics

Criteria	Metric	Performance profile	Data profile criterion	Accuracy
Capacity	$\mathrm{C1_R}$	$\mathrm{C1_R}$	$\mathrm{C1_R} \geq (1-\varepsilon_{\mathrm{C1_R}})/\lvert\mathcal{T}\rvert$	$\varepsilon_{\mathrm{C1_R}} = 0.5000$
Convergence-diversity	NHV	NHV	$\mathrm{NHV} \geq 1-\varepsilon_{\mathrm{NHV}}$	$\varepsilon_{\mathrm{NHV}} = 0.0075$
Combined	$\mathrm{C1_R}$+NHV	$\mathrm{C1_R}$ + NHV	$\mathrm{C1_R} + \mathrm{NHV} \geq (1-\varepsilon_{\mathrm{NHV}}) + (1-\varepsilon_{\mathrm{C1_R}})/\lvert\mathcal{T}\rvert$	$\varepsilon_{\mathrm{C1_R}} = 0.5000$ $\varepsilon_{\mathrm{NHV}} = 0.0075$

8.1. Analytical test problems

Figure 4 shows the relative performance variability of the algorithm formulation/setup parameters. Considering $C1_R$, the initialization and the formulation are the most significant parameters for a low and high budget of problem evaluations, respectively. Considering NHV, the algorithm formulation is the most significant parameter. Considering the sum of both metrics, the initialization has a decreasing significance as the number of problem evaluations increases, whereas the formulation shows the opposite trend. Overall, the coefficient set is the least relevant parameter.



(a) $C1_R$ (b) NHV (c) $C1_R + \text{NHV}$

Figure 4: Analytical test problems, relative performance variability of formulation/parameters

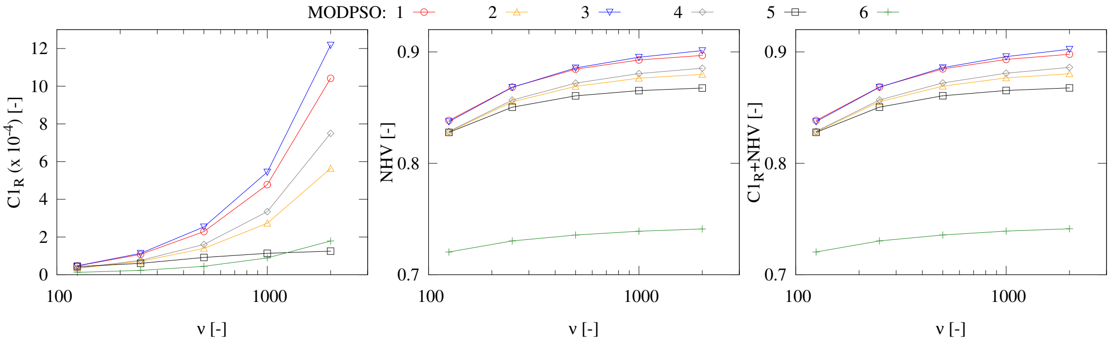
Figure 5 shows the average performance conditional to (a) the algorithm formulation, (b) the number of particles, (c) the particle initialization, and (d) the coefficient set.

Figure 5a shows that MODPSO1 and MODPSO3 have the most promising performances overall, whereas MODPSO6 is found as the least performing formulation. The formulations that use the variable domain in order to identify the cognitive and the social attractors (MODPSO1 and MODPSO3) have similar performances. Similarly, the formulations that use the codomain (MODPSO2 and MODPSO4) also show very close trends with in general a worse performance.

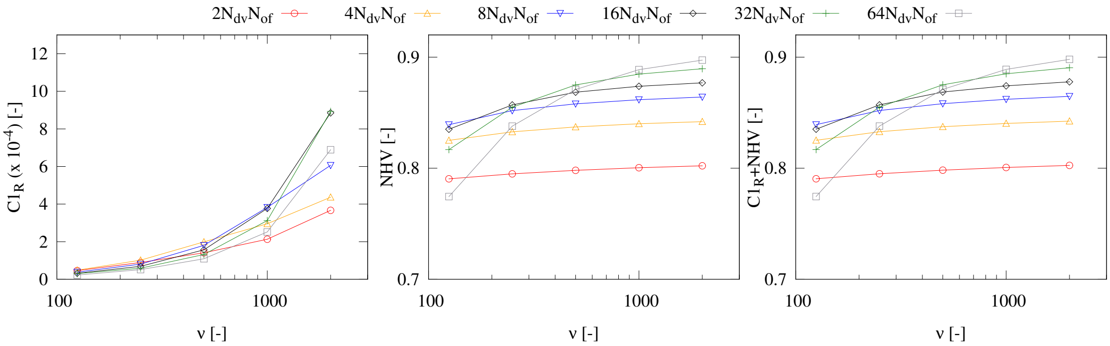
Figure 5b shows how using a large number of particles provides good performances for a high budget of problem evaluations. A large number of particles with a small budget of problem evaluations implies few algorithm iterations, therefore a reduced exploration and exploitation of the research space.

Figure 5c shows that the use of non-null velocity initialization has better performances than null-velocity initialization. The initialization with the particles on boundaries domain only (B) has good performances in terms of $C1_R$, whereas the initializations with points inside the domain (A and C) show good performances in terms of NHV and $C1_R + \text{NHV}$.

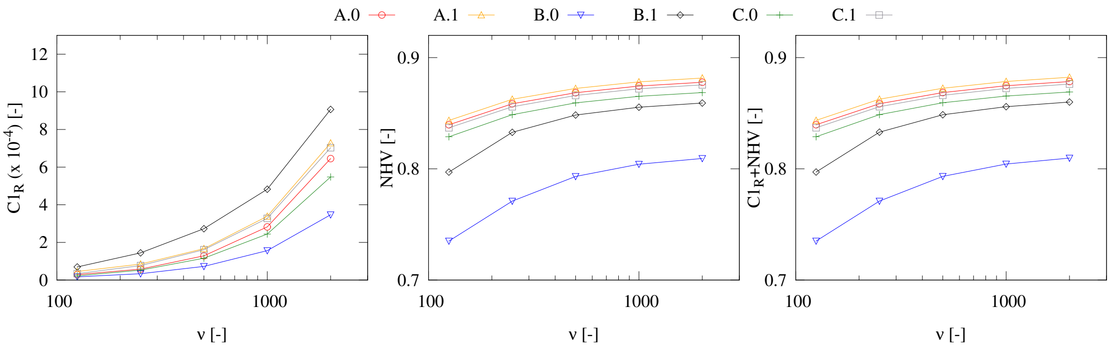
Finally, the most effective coefficient sets (on average) are set 5 [37] for $C1_R$ and set 4 [9] for NHV and $C1_R + \text{NHV}$, as shown in Fig. 5d.



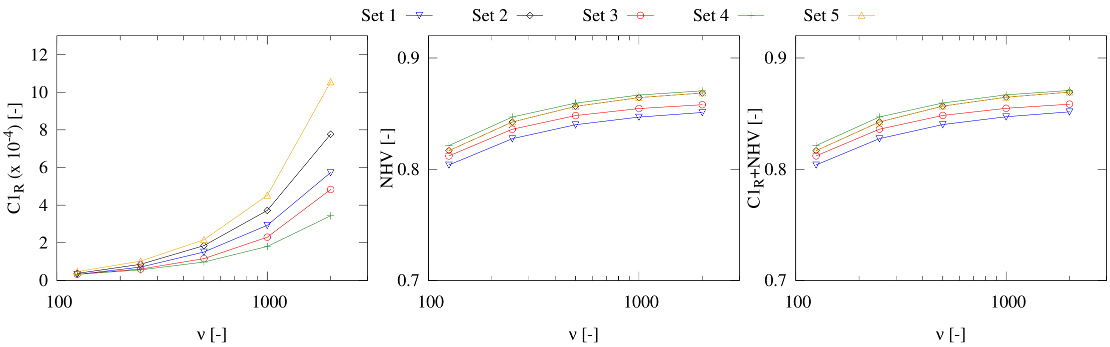
(a) Average performances, conditional to algorithm formulation



(b) Average performances, conditional to number of particles



(c) Average performances, conditional to particles initialization



(d) Average performances, conditional to coefficient set

Figure 5: Analytical test problems

Table 6 collects the most promising parameter setup (based on ν-averaged $\mathrm{C1_R}$+NHV) for each MODPSO formulation. Average values and standard deviations (STD) overall are also provided. The most promising formulation/parameter setup is MODPSO3 with $N_\mathrm{p} = 8N_\mathrm{of}N_\mathrm{dv}$, initialized over the domain and boundary with non-null velocity (C.1), and the coefficients proposed in [9] ($\chi = 0.721$, $c_1 = c_2 = 1.655$). Figure 6 shows the performances of this MODPSO3 setup, compared to average values, standard deviations, and the best performing formulation/parameters ("Best") for each specific budget ν. The algorithm proposed is very close to the "Best" values, especially considering NHV and $\mathrm{C1_R} + \mathrm{NHV}$.

Table 6: Most promising parameter setup for each MODPSO formulation, based on $\mathrm{C1_R} + \mathrm{NHV}$

ν	Form.	$\frac{N_\mathrm{p}}{N_\mathrm{of}N_\mathrm{dv}}$	Init.	Coef.	$\mathrm{C1_R}$	NHV	$\mathrm{C1_R} + \mathrm{NHV}$
Average	1	8	C.1	4	5.7135E-4	9.2101E-1	9.2158E-1
	2	8	B.1	2	9.6060E-4	9.0833E-1	9.0929E-1
	3	8	C.1	4	6.2921E-4	9.2334E-1	**9.2397E-1**
	4	8	B.1	2	1.2029E-3	9.1104E-1	9.1225E-1
	5	64	A.1	2	2.8450E-5	8.2530E-1	8.2533E-1
	6	8	A.0	5	7.6128E-5	9.0520E-1	9.0527E-1
	Average				2.4321E-4	8.4464E-1	8.4488E-1
	STD				4.2039E-4	7.4424E-2	7.4579E-2

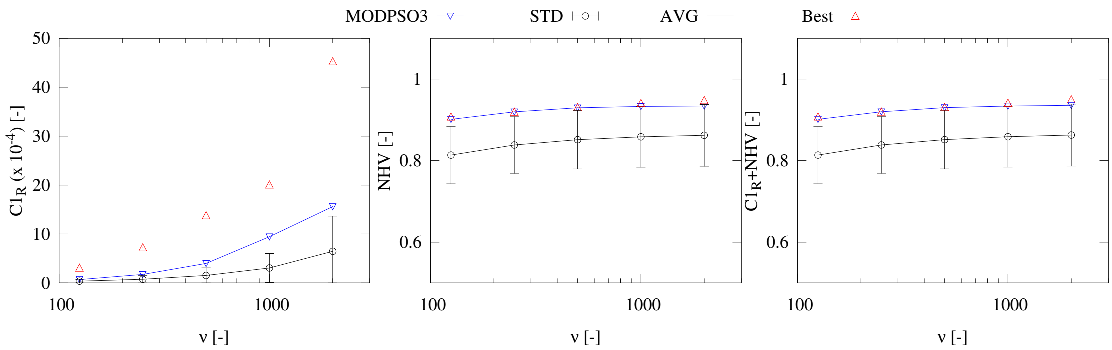
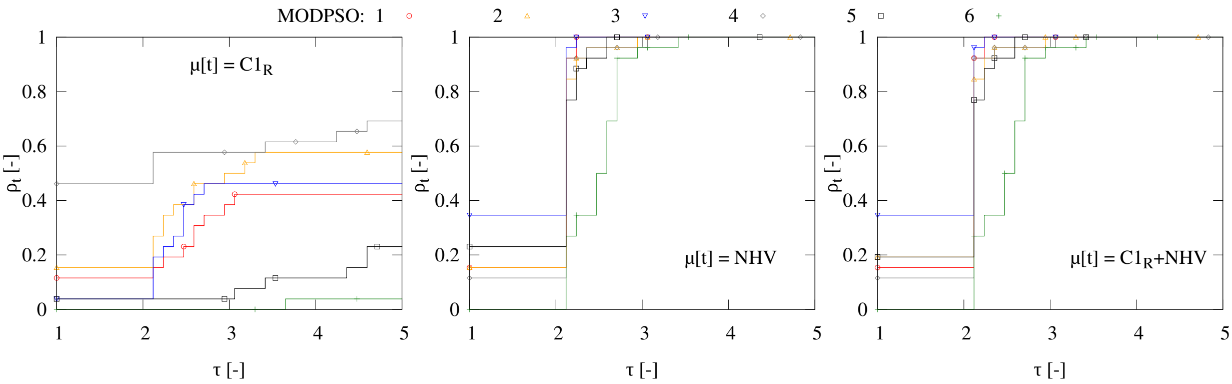
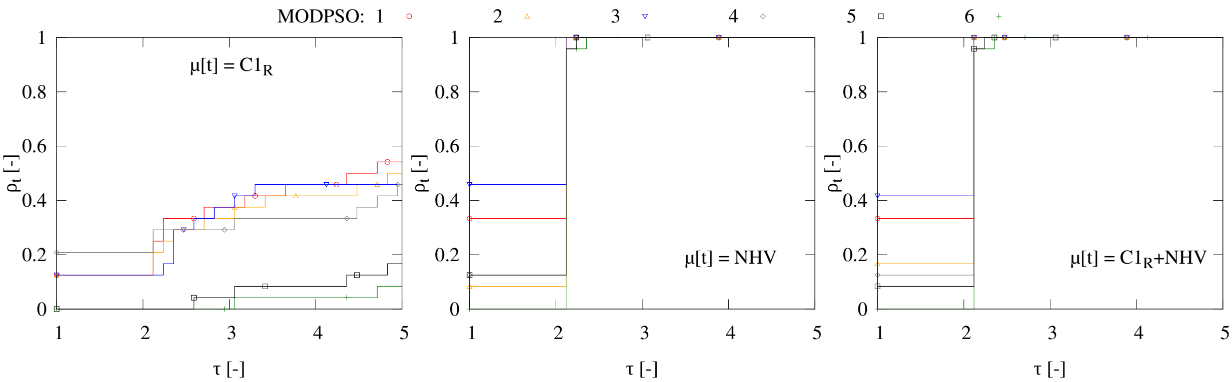


Figure 6: Analytical test problems, most promising formulation/setup (MODPSO3) performances compared to average values, standard deviation, and "Best" implementations

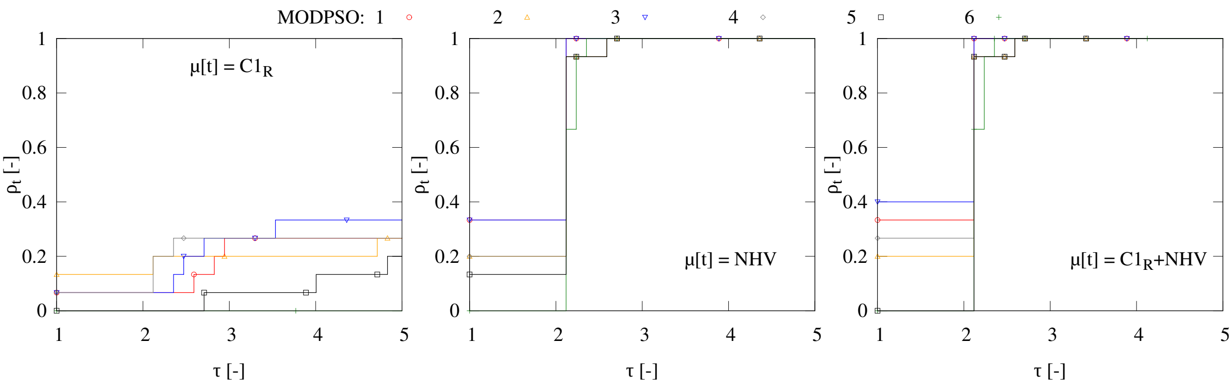
The best performing setup for each MODPSO formulation (Tab. 6) has been assessed in terms of performance profiles, conditional to the shape of the Pareto front (convex/concave and continuous/discontinuous, see Fig. 7) and its dimension (2D and 3D, see Fig. 8), for an average budget of problem evaluations equal to $500N_\mathrm{of}N_\mathrm{dv}$. It is worth noting that, for $\mathrm{C1_R}$, the unity of the performance profile is never achieved, since none of the algorithms considered is able to provide reference solutions for all the test problems. Performance and data profiles confirm that MODPSO3 is the most promising algorithm in terms of NHV and $\mathrm{C1_R} + \mathrm{NHV}$ (see Figs. 7-9). Data profiles (Fig. 9b) show that MODPSO3 is the best performing algorithm for all the evaluation budgets, on average.



(a) Continuous concave Pareto front

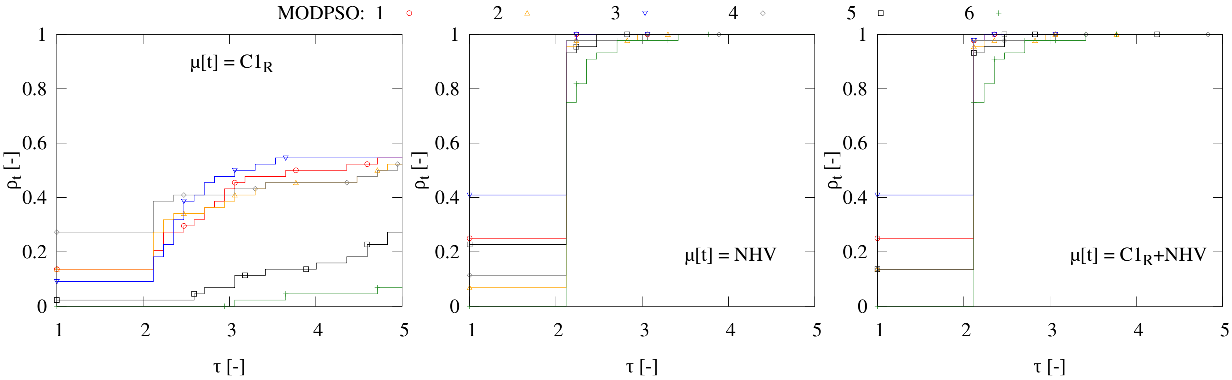


(b) Continuous convex Pareto front

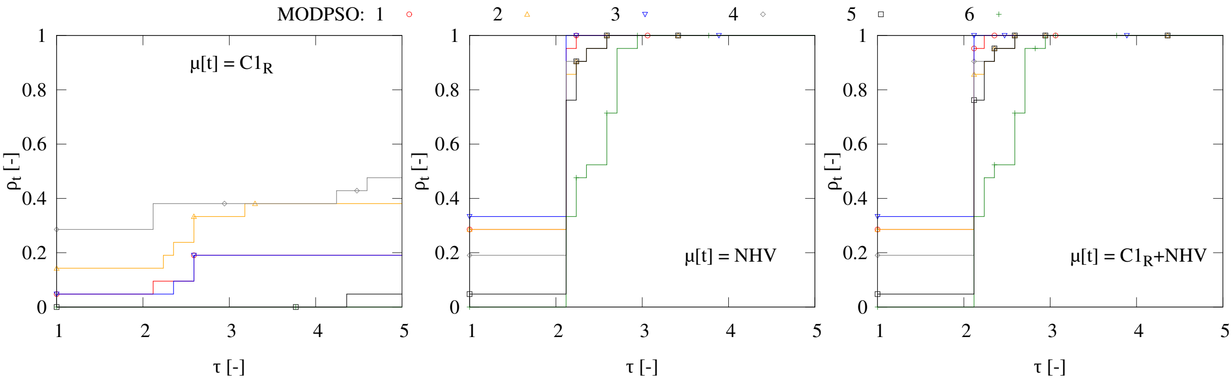


(c) Discontinuous convex/concave Pareto front

Figure 7: Performance profiles of the most promising setup of each formulation for $500 N_{\rm of} N_{\rm dv}$ problem evaluations, conditional to the shape of the Pareto front

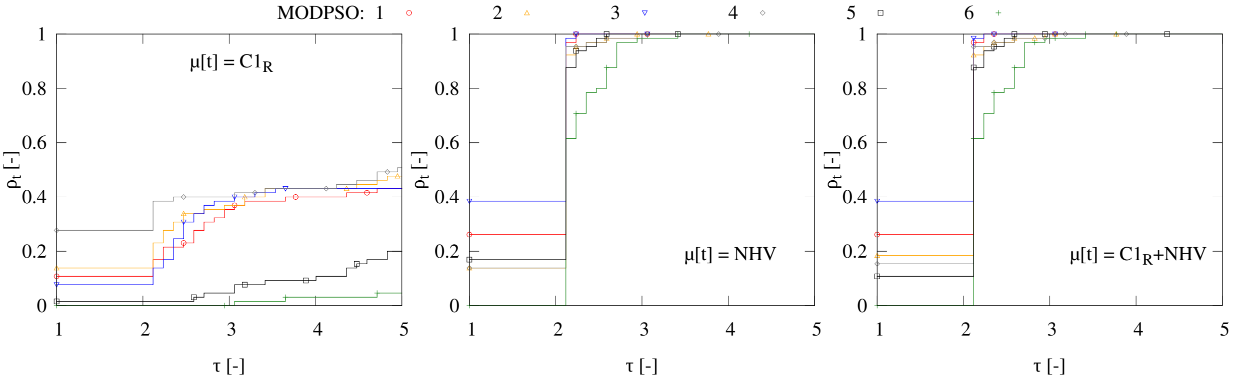


(a) 2D Pareto front

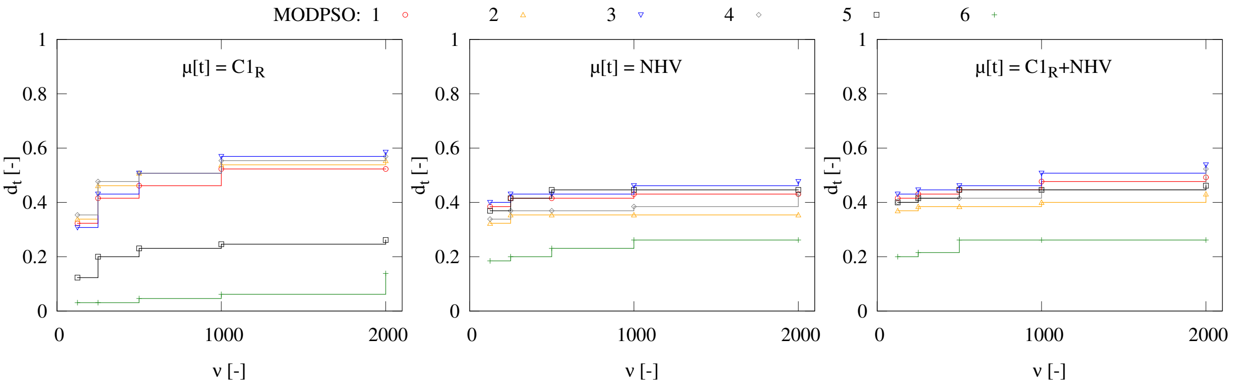


(b) 3D Pareto front

Figure 8: Performance profiles of the most promising setup of each formulation for $500N_{\rm of}N_{\rm dv}$ functions evaluations, conditional to the dimension of the Pareto front



(a) Performance profiles



(b) Data profiles

Figure 9: Performance and data profiles of the most promising setup of each formulation for $500N_{\rm of}N_{\rm dv}$, conditional to the whole set of problems

8.2. Catamaran hull-form optimization

The optimization is performed with MODPSO3 from Tab. 6. This algorithm is compared with its random version, hereafter called MOPSO3 implemented following Eq. 7. In order to provide a comparison with other swarm-intelligence and evolutionary algorithms from literature, the optimization is performed also with MOPSO-CRM, MOPSO-CD, and NSGA-II with the parameters suggested in [22], [17], and [18], respectively. A budget of problem evaluations equal to $2000N_{\rm of}N_{\rm dv}$ is used for all algorithms.

Figure 10 shows the algorithms performance versus the number of problem evaluations. The results of the stochastic algorithms are based on 100 optimizations: the solid line represent the median performance, whereas the color band represent the 95%-confidence interval. Considering $\rm C1_R$, MOPSO-CD shows the best results for low budgets of problem evaluations, without achieving any significant improvement as the available budget increases. This is reasonable if one considers the finite-size archive of non-dominated solutions used by the method. MODPSO3 shows comparable performances with NSGA-II, outperforming its random version (MOPSO3) for medium/high budgets, and achieving close results to MOPSO-CD for NHV and $\rm C1_R + NHV$ for high budgets. Finally, MODPSO-CRM shows a quite slow convergence compared to MODPSO3 and the other algorithms and is not able to provide any reference points. Nevertheless, its trend is monotonically convergent to high values of NHV achieving a significant reduction of its confidence band.

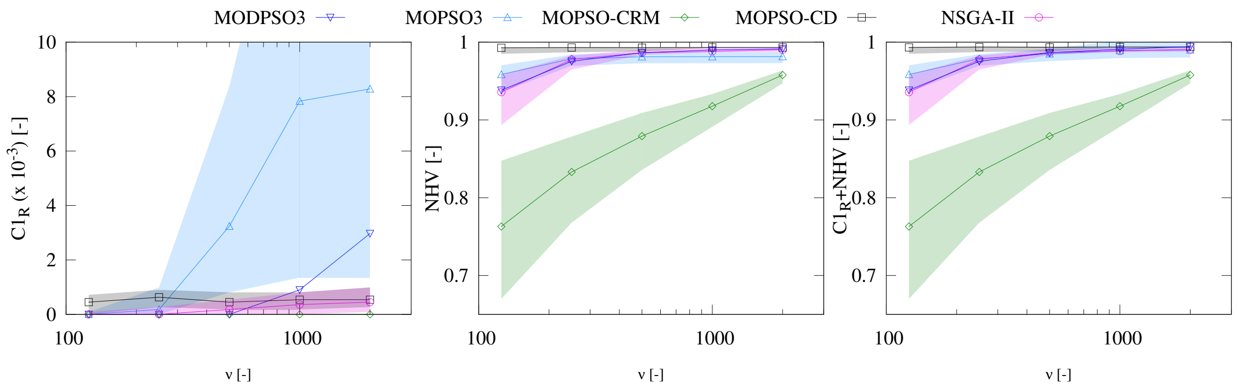
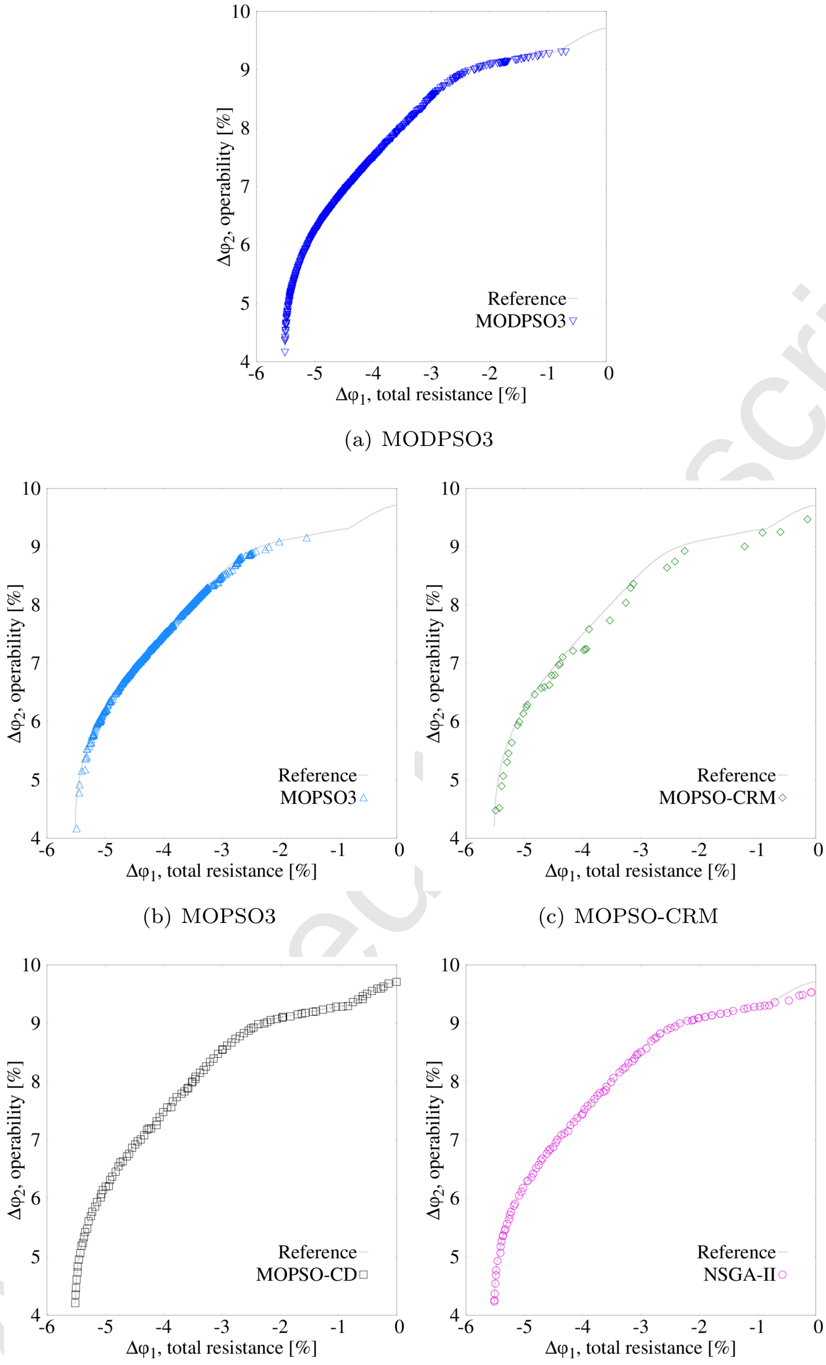


Figure 10: Algorithms performance convergence for the catamaran problem



(a) MODPSO3

(b) MOPSO3

(c) MOPSO-CRM

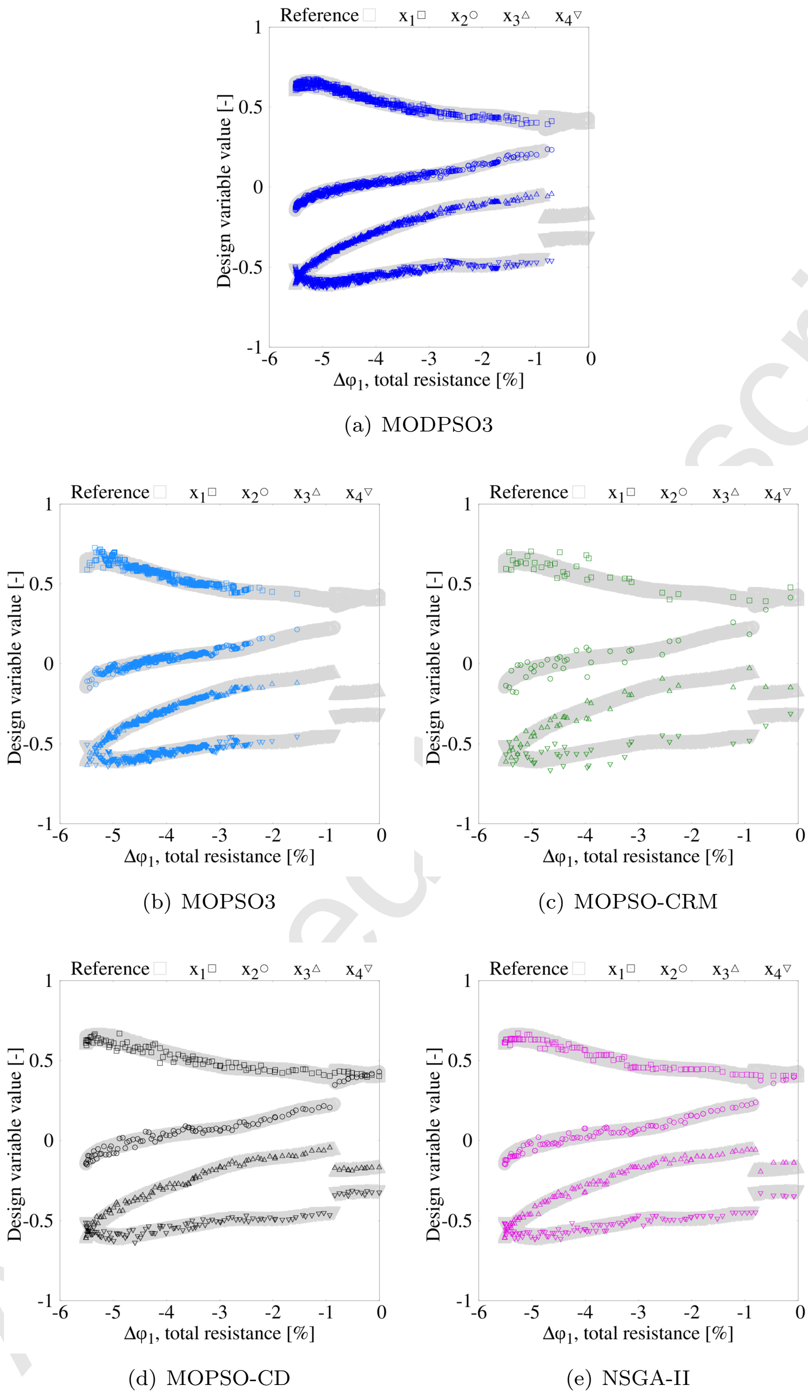
(d) MOPSO-CD

(e) NSGA-II

Figure 11: Reference and solution sets for the catamaran hull-form optimization for $2000 N_{\rm dv} N_{\rm of}$ problem evaluations

Figure 11 compares the algorithms solutions to the reference set. The solution associated to the median performance is shown for the stochastic algorithms. MODPSO3 shows a better solution set compared to its random version (MOPSO3), although is unable to cover the upper right part of the front. MOPSO-CRM provides a solution set with fewer and less accurate points than the other algorithms. Nevertheless, it is able to explore the upper right part of the reference set more effectively than MODPSO3 and MOPSO3.

MOPSO-CD and NSGA-II cover the whole reference set, with a good distribution of their solutions. NSGA-II is not able to reach the reference set completely in upper right part of the front. Comparing MODPSO3 and MOPSO-CD it can be said that, for this specific problem, MOPSO-CD covers the whole reference set, whereas MODPSO3 finds a larger number of solutions in the middle region of the front, representing the most interesting zone from an engineering viewpoint. Figure 12 shows the design variable values in $\mathcal{S}$ versus the first objective function. It is worth noting that, consistently to Fig. 11, MOPSO-CD is the most effective in reaching the reference solution characterized by a discontinuity in the design variable space.



(a) MODPSO3

(b) MOPSO3

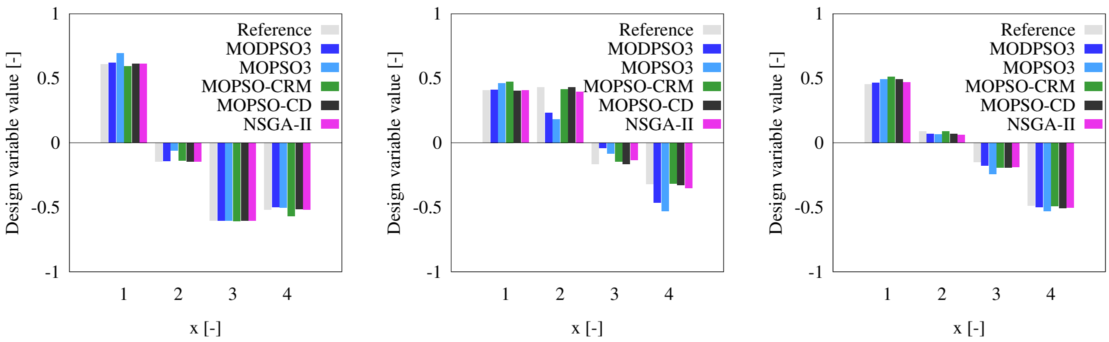
(c) MOPSO-CRM

(d) MOPSO-CD

(e) NSGA-II

Figure 12: Variables values for reference solution and solution sets for the catamaran hull-form optimization for $2000N_{\rm dv}N_{\rm of}$ problem evaluations

Figure 13 and Tab. 7 compare algorithms and the reference solution, showing the design variables for three different configurations corresponding to: (a) minimum expected value of the mean total resistance, (b) maximum ship operability, and (c) minimum aggregate objective function (with equal weights). The largest differences in the design variables are found for $\varphi_{2,\max}$ (see Fig. 13b). This depends on the algorithms ability of identifying the upper right part of the reference solution. The corresponding optimal hulls are finally compared to the original in Fig. 14.

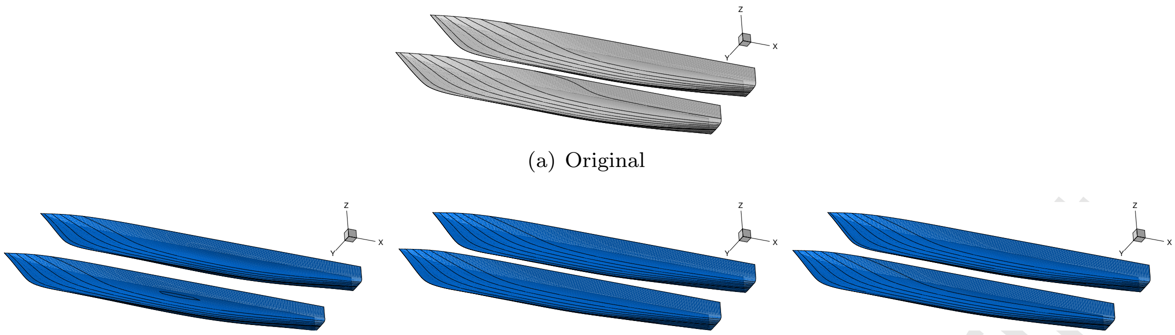


(a) Minimum resistance, $\varphi_{1,\min}$ (b) Maximum operability, $\varphi_{2,\max}$ (c) Best trade-off, $\min(\varphi_1 - \varphi_2)$

Figure 13: Hull-form optimization, optimal design variable values for three Pareto solutions

Table 7: Hull-form optimization problem results

Configurations	Algorithm	x_1	x_2	x_3	x_4	$\Delta\varphi_1\%$	$\Delta\varphi_2\%$
$\varphi_{1,\min}$	Reference	0.804	0.427	0.198	0.242	-5.514	4.207
	MODPSO3	0.810	0.430	0.198	0.250	-5.512	4.158
	MOPSO3	0.847	0.470	0.197	0.249	-5.433	4.680
	MOPSO-CRM	0.798	0.432	0.196	0.215	-5.489	4.476
	MOPSO-CD	0.805	0.428	0.198	0.242	-5.514	4.207
	NSGA-II	0.807	0.427	0.198	0.240	-5.514	4.232
$\varphi_{2,\max}$	Reference	0.703	0.716	0.418	0.341	-0.001	9.710
	MODPSO3	0.706	0.617	0.480	0.268	-0.697	9.303
	MOPSO3	0.730	0.592	0.458	0.235	-1.395	9.087
	MOPSO-CRM	0.737	0.708	0.428	0.343	-0.143	9.467
	MOPSO-CD	0.702	0.715	0.418	0.336	-0.004	9.708
	NSGA-II	0.703	0.698	0.433	0.324	-0.079	9.529
$\min(\varphi_1 - \varphi_2)$	Reference	0.727	0.545	0.425	0.255	-2.757	8.763
	MODPSO3	0.733	0.534	0.412	0.251	-3.084	8.460
	MOPSO3	0.745	0.532	0.379	0.235	-3.537	7.976
	MOPSO-CRM	0.756	0.544	0.404	0.255	-3.130	8.360
	MOPSO-CD	0.745	0.534	0.405	0.248	-3.205	8.339
	NSGA-II	0.734	0.530	0.406	0.248	-3.216	8.318



(a) Original

(b) Minimum resistance (c) Maximum operability (d) Best trade-off

Figure 14: MODPSO3 optimal hull compared to the original

9. Conclusions

A systematic assessment and selection have been presented and discussed of the most promising formulation and setup parameters for multi-objective deterministic particle swarm optimization in simulation-based problems. A comparative study of six formulations (varying the definition of cognitive and social attractors) and three setting parameters (number of particles, initialization, and coefficient set) have been performed using 66 analytical test problems, with a number of objective functions ranging from two to three and a number of variables from two to eight (as often encountered in simulation-based optimization). The problems have been presented setting apart continuous/discontinous and convex/non-convex Pareto fronts. A full-factorial combination of formulations and parameters has been investigated through more than 60,000 optimization runs and three performance metrics ($C1_R$, NHV, and their sum).

The algorithm formulation has a significant effect on the performance, greater than the setup parameters. Specifically, MODPSO3 has been found with the most promising performance, for all types of Pareto front. The formulations that use the variable domain for the attractor identification (MODPSO1 and MODPSO3) have similar performances. The formulations that embed the attractor identification in the codomain (MODPSO2 and MODPSO4) also show very close trends, with worse performance than MODPSO1 and MODPSO3.

Using a large number of particles is shown to have good performances for a high budget of problem evaluations, whereas a small number of particle should be preferred if the available budget of problem evaluations is small. Non-null velocity initialization has shown good overall performances and shall be preferred. Moreover, initializing the particles inside the domain have shown better performances than the initialization on the domain boundary only. Finally, the coefficient sets by [7] and [9] show the best overall performance. The most promising setups for each formulation have been summarized in Tab. 6 and MODPSO3 (with a number of particles equal to $8N_{\rm of}N_{\rm dv}$, initialized over the domain and boundary with non-null velocity, and the coefficients corresponding to $\chi = 0.721$, $c_1 = c_2 = 1.655$ [9]) has been selected as the most promising algorithm overall.

MODPSO3 have been used for the hull-form optimization of a high-speed catamaran in realistic ocean conditions, addressing the expected value of the mean total resistance and the ship operability in irregular waves. A metamodel has been used, based on URANS simulations. Four stochastic optimization algorithms (MOPSO3, MOPSO-CRM, MOPSO-CD, and NSGA-II) have been applied and compared with MODPSO3. The latter has shown the capability of achieving comparable results to the stochastic algorithms.

It may be noted that often the best performing setting parameters provided by $C1_R$ are completely different from those provided by NHV. This confirms the different point of views offered by the two metrics and that considering both $C1_R$ and NHV provides a robust analysis.

A final consideration can be made about the efficiency and effectiveness of MOPSO-CD, compared to other MODPSOs. Particle velocities are easily found greater than for other MOPSO algorithms. This stems from using a limited-size set of non-dominated solutions with large crowding distance. Since non-dominated solution set extremes are always attractors (based on their crowding distance) the algorithm easily tends to extend the exploration of the non-dominated solution set, with beneficial effects on the NHV. This is particularly evident for the catamaran problem. Moreover, MODPSO3 is able to achieve the same NHV as MOPSO-CD, for a high budget of problem evaluations, although is unable to cover the entire front.

Based on the present results, future work will include the development and assessment of hybrid extensions of MODPSO with derivative-free local multi-objective algorithms, in order to improve the exploration and exploitation of the research space. Further deterministic formulations will be evaluated, based on the crowding distance.

Acknowledgements

The present research is supported by the US Office of Naval Research, NICOP Grant N62909-15-1-2016, under the administration of Dr. Woei-Min Lin and by the Italian Flagship Project RITMARE, coordinated by the Italian National Research Council and funded by the Italian Ministry of Education.

References

[1] A. Pinto, D. Peri, E. F. Campana, Global optimization algorithms in naval hydrodynamics, Ship Technology Research 51 (3) (2004) 123–133.
[2] T. W. Simpson, J. Poplinski, P. N. Koch, J. K. Allen, Metamodels for computer-based engineering design: survey and recommendations, Engineering with computers 17 (2) (2001) 129–150.
[3] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. K. Tucker, Surrogate-based analysis and optimization, Progress in aerospace sciences 41 (1) (2005) 1–28.
[4] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Neural Networks, 1995. Proceedings, IEEE International Conference on, Vol. 4, 1995, pp. 1942–1948.
[5] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, Vol. 3, IEEE, 1999.
[6] Y. Shi, R. Eberhart, Parameter selection in particle swarm optimization, in: Evolutionary programming VII, Springer, 1998, pp. 591–600.
[7] I. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information processing letters 85 (6) (2003) 317–325.
[8] R. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, Vol. 1, IEEE, 2000, pp. 84–88.
[9] M. Clerc, Stagnation analysis in particle swarm optimization or what happens when nothing happens, Online at http://clerc. maurice. free. fr/pso.
[10] J. Chen, Y. Tang, X. Huang, Application of surrogate based particle swarm optimization to the reliability-based robust design of composite pressure vessels, Acta Mechanica Solida Sinica 26 (5) (2013) 480 – 490.
[11] P. N. Kechagiopoulos, G. N. Beligiannis, Solving the urban transit routing problem using a particle swarm optimization based algorithm, Applied Soft Computing 21 (2014) 654 – 676.
[12] 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 59 (2016) 115 – 128.
[13] J. Moore, R. Chapman, Application of particle swarm to multiobjective optimization, Department of Computer Science and Software Engineering, Auburn University.
[14] C. C. Coello, M. S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, Vol. 2, IEEE, 2002, pp. 1051–1056.
[15] J. E. Fieldsend, S. Singh, A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence, in: 2002 UK Workshop on Computational Intelligence, pp. 37 - 44, Birmingham, UK, 2-4 September 2002, 2002.
[16] S. Mostaghim, J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO), in: Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE, IEEE, 2003, pp. 26–33.
[17] C. R. Raquel, P. C. Naval Jr, An effective use of crowding distance in multiobjective particle swarm optimization, in: Proceedings of the 7th annual conference on Genetic and evolutionary computation, ACM, 2005, pp. 257–264.
[18] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE transactions on evolutionary computation 6 (2) (2002) 182–197.
[19] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, E. Alba, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, in: Computational intelligence in multi-criteria decision-making, 2009. MCDM'09. IEEE symposium on, IEEE, 2009, pp. 66–73.
[20] I. C. García, C. A. C. Coello, A. Arias-Montano, MOPSOhv: a new hypervolume-based multi-objective particle swarm optimizer, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 266–273.
[21] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms - a comparative case study, in: Parallel problem solving from Nature - PPSN V, Springer, 1998, pp. 292–301.

[22] Y.-J. Zheng, H.-F. Ling, J.-Y. Xue, S.-Y. Chen, Population classification in fire evacuation: A multiobjective particle swarm optimization approach, IEEE Transactions on Evolutionary Computation 18 (1) (2014) 70–81.
[23] W. Hu, G. G. Yen, Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system, IEEE Transactions on Evolutionary Computation 19 (1) (2015) 1–18.
[24] W. Hu, G. G. Yen, Density estimation for selecting leaders and mantaining archive in MOPSO, in: Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE, 2013, pp. 181–188.
[25] K. Parsopoulos, D. Tasoulis, M. Vrahatis, Multiobjective optimization using parallel vector evaluated particle swarm optimization, in: Proceedings of the IASTED international conference on artificial intelligence and applications (AIA 2004), Vol. 2, 2004, pp. 823–828.
[26] J. D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: Proceedings of the 1st international Conference on Genetic Algorithms, L. Erlbaum Associates Inc., 1985, pp. 93–100.
[27] W. Peng, Q. Zhang, A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems, in: Granular Computing, 2008. GrC 2008. IEEE International Conference on, IEEE, 2008, pp. 534–537.
[28] M. Reyes-Sierra, C. Coello Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art, International journal of computational intelligence research 2 (3) (2006) 287–308.
[29] S. Lalwani, S. Singhal, R. Kumar, N. Gupta, A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm, Transactions on Combinatorics 2 (1) (2013) 39–101.
[30] Y. Guo, N. Li, H. Zhang, T. Ye, Elitist vector evaluated particle swarm optimization for multi-mode resource leveling problems, Journal of Computational Information Systems 8 (2012) 3697–3705.
[31] A. M. Baltar, D. G. Fontane, A generalized multiobjective particle swarm optimization solver for spreadsheet models: application to water quality, Proceedings of the twenty sixth annual American geophysical union hydrology days (2006) 20–22.
[32] A. M. Sharaf, A. A. El-Gammal, A novel discrete multi-objective particle swarm optimization (MOPSO) of optimal shunt power filter, in: Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES, IEEE, 2009, pp. 1–7.
[33] Y. Liu, Automatic calibration of a rainfall–runoff model using a fast and elitist multi-objective particle swarm algorithm, Expert Systems with Applications 36 (5) (2009) 9533–9538.
[34] M. Diez, E. F. Campana, F. Stern, Development and evaluation of hull-form stochastic optimization methods for resistance and operability, in: Proceedings of the 13th International Conference on Fast Sea Transportation, FAST 2015, Washington, D.C., USA, 2015.
[35] A. Serani, C. Leotardi, U. Iemma, E. F. Campana, G. Fasano, M. Diez, Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems, Applied Soft Computing 49 (2016) 313 – 334.
[36] A. Pinto, D. Peri, E. F. Campana, Multiobjective optimization of a containership using deterministic particle swarm optimization, Journal of Ship Research 51 (3) (2007) 217–228.
[37] M. Diez, D. Peri, Robust optimization for ship conceptual design, Ocean Engineering 37 (11) (2010) 966–977.
[38] 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.
[39] A. Serani, M. Diez, Are random coefficients needed in particle swarm optimization for simulation-based ship design?, in: Proceedings of the 7th International Conference on Computational Methods in Marine Engineering (Marine 2017), 2017.
[40] R. Pellegrini, E. F. Campana, M. Diez, A. Serani, F. Rinaldi, G. Fasano, U. Iemma, G. Liuzzi, S. Lucidi, F. Stern, Application of derivative-free multi-objective algorithms to reliability-based robust design optimization of a high-speed catamaran in real ocean environment, in: Engineering Optimization IV - Rodrigues et al. (Eds.), 2014.
[41] C. Hwang, A. Md. Masud, A simple multi-objective optimization problem, Vol. 164, Lecture Notes in Economics and Mathematical Systems, 1979, Ch. Multiple Objective Decision Making - Methods and Applications, p. 281.
[42] F. Kursawe, A variant of evolution strategies for vector optimization, in: Parallel Problem Solving from Nature, Springer, 1991, pp. 193–197.
[43] C. Fonseca, P. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part I: A unified formulation, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 28 (1) (1998) 26–37.
[44] F. Cheng, X. Li, Generalized center method for multiobjective engineering optimization, Engineering Optimization 31 (5) (1999) 641–661.
[45] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, Evolutionary computation 7 (3) (1999) 205–230.
[46] Y. Jin, M. Olhofer, B. Sendhoff, Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How?, in: Proceeding of GECCO, Genetic and Evolutionary Computation Conference, Morgan Kaufmann, 2001, pp. 1042–1049.
[47] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the Congress on Evolutionary Computation (CEC-2002), Honolulu, USA, 2002, pp. 825–830.
[48] T. Okabe, Y. Jin, M. Olhofer, B. Sendhoff, On test functions for evolutionary multi-objective optimization, in: Parallel Problem Solving from Nature-PPSN VIII, Springer, 2004, pp. 792–802.
[49] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in: Evolutionary multi-criterion optimization, Springer, 2005, pp. 280–295.
[50] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, Evolutionary Computation, IEEE Transactions on 10 (5) (2006) 477–506.

[51] A. Lovison, A synthetic approach to multiobjective optimization, arXiv preprint arXiv:1002.0093.
[52] E. F. Campana, A. Pinto, A multiobjective particle swarm optimization algorithm based on sub-swarms search, Tech. rep., CNR-INSEAN (2005).
[53] T. Wong, W. Luk, P. Heng, Sampling with hammersley and halton points, Journal of graphics tools 2 (2) (1997) 9–24.
[54] S. Jiang, Y.-S. Ong, J. Zhang, L. Feng, Consistencies and contradictions of performance metrics in multiobjective optimization, Cybernetics, IEEE Transactions on 44 (12) (2014) 2391–2404.
[55] P. Czyzak, A. Jaszkiewicz, Pareto simulated annealinga metaheuristic technique for multiple-objective combinatorial optimization, Journal of Multi-Criteria Decision Analysis 7 (1) (1998) 34–47.
[56] 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.
[57] 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.
[58] M. Diez, A. Serani, C. Leotardi, E. F. Campana, D. Peri, U. Iemma, G. Fasano, S. Giove, A Proposal of PSO Particles' Initialization for Costly Unconstrained Optimization Problems: ORTHOinit, Springer International Publishing, Cham, 2014, pp. 126–133.
[59] 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.
[60] E. Dolan, J. Moré, Benchmarking optimization software with performance profiles, Mathematical programming 91 (2) (2002) 201–213.
[61] A. Custódio, J. A. Madeira, A. Vaz, L. Vicente, Direct multisearch for multiobjective optimization, SIAM Journal on Optimization 21 (3) (2011) 1109–1140.
[62] J. Moré, S. Wild, Benchmarking derivative-free optimization algorithms, SIAM Journal on Optimization 20 (1) (2009) 172–191.
[63] W. He, M. Diez, Z. Zou, E. F. Campana, F. Stern, URANS study of Delft catamaran total/added resistance, motions and slamming loads in head sea including irregular wave and uncertainty quantification for variable regular wave and geometry, Ocean Engineering 74 (2013) 189 – 217.
[64] C. M. Fonseca, L. Paquete, M. Lòpez-Ibàñez, An improved dimension - sweep algorithm for the hypervolume indicator, in: In Proceedings of the 2006 Congress on Evolutionary Computation (CEC'06), IEEE Press, Piscataway, NJ, 2006, pp. 1157–1163.