Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems

Andrea Serani[1,2], Cecilia Leotardi[1], Umberto Iemma[2],
Emilio F. Campana[1], Giovanni Fasano[3], Matteo Diez[1,*]

[1]CNR-INSEAN, Natl. Research Council–Marine Technology Research Inst., Rome, Italy
[2]Department of Mechanical and Industrial Engineering, Roma Tre University, Rome, Italy
[3]Department of Management, Ca' Foscari University of Venice, Venice, Italy
[*]Corresponding author. Email: matteo.diez@cnr.it

Post-print

Applied Soft Computing
DOI: 10.1016/j.asoc.2016.08.028
Accepted: 12 August 2016
The final publication is available at Elsevier via
http://dx.doi.org/10.1016/j.asoc.2016.08.028

Abstract

Deterministic optimization algorithms are very attractive when the objective function is computationally expensive and therefore the statistical analysis of the optimization outcomes becomes too expensive. Among deterministic methods, deterministic particle swarm optimization (DPSO) has several attractive characteristics such as the simplicity of the heuristics, the ease of implementation, and its often fairly remarkable effectiveness. The performances of DPSO depend on four main setting parameters: the number of swarm particles, their initialization, the set of coefficients defining the swarm behavior, and (for box-constrained optimization) the method to handle the box constraints. Here, a parametric study of DPSO is presented, with application to simulation-based design in ship hydrodynamics. The objective is the identification of the most promising setup for both synchronous and asynchronous implementations of DPSO. The analysis is performed under the assumption of limited computational resources and large computational burden of the objective function evaluation. The analysis is conducted using 100 analytical test functions (with dimensionality from two to fifty) and three performance criteria, varying the swarm size, initialization, coefficients, and the method for the box constraints, resulting in more than 40,000 optimizations. The most promising setup is applied to the hull-form optimization of a high speed catamaran, for resistance reduction in calm water and at fixed speed, using a potential-flow solver.

Keywords: Simulation-based design; derivative-free optimization; global optimization; particle swarm optimization; ship hydrodynamics optimization

1 Introduction

Particle Swarm Optimization (PSO) was originally introduced in [1], 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 heuristic algorithms for single-objective evolutionary derivative-free global optimization. Derivative-free global optimization approaches are often preferred to local approaches when objectives are nonconvex and/or noisy, and when multiple local optima cannot be excluded, as often encountered in simulation-based design (SBD) optimization. The computational burden of global optimization techniques is usually much larger compared to local methods, so that the accuracy of the solution sought often depends on the available computational resources.

Zhang et al. [2] presents a comprehensive survey on the PSO variants and their application in several engineering fields, such as mechanical or chemical. Recent applications of PSO to ship SBD include medium- to high-fidelity hull-form and waterjet design optimization of fast catamarans, by morphing techniques [3, 4] and geometry modifications based on Karhunen-Loève expansion (KLE) [5, 6, 7], and low- to medium-fidelity optimization of unconventional multi-hull configurations [8]. When global techniques are used in design optimization, 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 [9, 10], 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 procedures are available.

The original PSO makes use of random coefficients, aiming at sustaining the variety of the swarm 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 computer simulations are used directly as analysis tools. Furthermore, if the design problem in hand is scheduled within an accurate project planning, time resources might be a tight bound for the optimization process. 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 [11] and random PSO [5]. Moreover, the availability of parallel architectures and high performance computing (HPC) systems has offered the opportunity to extend the original *synchronous* implementation of PSO (SPSO) to CPU-time efficient *asynchronous* methods (APSO), assessed on test functions in [12], and applied in several engineering problems such as multidisciplinary optimization of commercial aircraft [13], biomechanics [14], and swarm robotics [15]. Using distributed computing, synchronous implementation of PSO are implemented when at iteration $k+1$ the position and velocity of any particle is updated after evaluating the function at *all* the particles positions at iteration k. In an asynchronous implementation of PSO the position and velocity of a particle is possibly based on the fitness value at *a subset of all* particles positions.

The effectiveness and efficiency of PSO 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 behaviour 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 [16, 17, 18]. More recently the effects of the particles initialization have been studied in [19, 20], the effects of the coefficients have been shown in

[21], whereas the bounds handling techniques have been presented in [22]. A comprehensive study on the PSO parameter selection has been presented in [23] and a preliminary assessment of the performances of DPSO, varying (a), (b) and (c), is presented in [24]. A survey of approaches for general constrained optimization problems in industrial design and multidisciplinary design optimization may be found in [25], including also general nonlinear constraints. However, the discussion on the application of DPSO in SBD problems is still limited, lacking a systematic and a comparative analysis.

The objective of the present work is the identification of the most effective and efficient parameters for both synchronous and asynchronous deterministic particle swarm optimization (SDPSO and ADPSO), for their use in SBD procedures. The focus is on industrial problems, directly using CPU-time expensive analyses. These make the statistical analysis of the results too expensive and therefore demand for deterministic algorithms. Due to the attractive features of DPSO (such as the simplicity of the heuristics, the ease of implementation, and its often fairly remarkable effectiveness in industrial problems), the current study is limited to DPSO and its implementations. A systematic comparison of DPSO with other deterministic and stochastic methods is beyond the scopes of the present work.

The approach includes a preliminary parametric analysis on 100 analytical test functions [26, 27, 28, 29] characterized by different degrees of non-linearities and number of local minima, with full-factorial combination of: (a) number of particles (using a power of two times the number of design variables); (b) initialization of the particles position and velocity (using Hammersley sequence sampling (HSS) [30]); (c) set of coefficients, chosen from literature [12, 16, 17, 31, 32]; (d) inelastic and semi-elastic wall-type approach for box constraints [22]. In order to handle the box constraints, wall-type approaches are preferred instead of penalty approaches or Lagrangian functions, which might introduce additional bias in the analysis. The number of design variables ranges from two to fifty and the simulation budget (maximum number of objective function evaluations) is up to 4096 times the number of design variables. The preliminary parametric analysis is conducted on two subsets of problems, respectively with less and more than ten design variables. A Intel Xeon E5-1620 v2 3.70GHz is used for the preliminary tests. Three absolute metrics are defined and applied for the evaluation of the algorithm performances, based on the distance between PSO-found solutions and analytical optima. According with the numerical tests, the most effective parameter choice among (a), (b), (c), and (d) is identified, based on the associated relative variability of the results. Then, the most promising setups for SDPSO and ADPSO are determined and applied to an industrial problem, namely a fast catamaran hull-form optimization, for calm water and fixed speed. The objective function is the ratio R_T/W between the total resistance (R_T) and the ship weight (W). The hull-form modification is performed using a KLE-based morphing approach [5, 6, 7], using respectively four- and six-dimensional design spaces. Computer simulations are performed using the potential flow (PF) solver CNR-INSEAN WARP [33], on a cluster of Intel Xeon E5462 2.80GHz. Each function evaluation takes about 10 minutes per node. Additionally, the optimization results are compared with those obtained in earlier research, based on a high-fidelity URANS solver [5].

2 PSO formulations

Consider the following *objective function*:

$$f(\mathbf{x}) \ : \ \mathbb{R}^{N_{dv}} \longrightarrow \mathbb{R} \tag{1}$$

and the global optimization problem

$$\min_{\mathbf{x}\in\mathcal{L}} f(\mathbf{x}), \quad \mathcal{L} \subset\mathbb{R}^{N_{dv}} \tag{2}$$

where $\mathcal{L}$ is a closed and bounded subset of $\mathbb{R}^{N_{dv}}$ and N_{dv} is the number of design variables. The global minimization of the objective function $f(\mathbf{x})$ requires to find a vector $\mathbf{a} \in \mathcal{L}$ such that:

$$\forall\, \mathbf{b} \in \mathcal{L} \;\; : \;\; f(\mathbf{a}) \leq f(\mathbf{b}) \tag{3}$$

Then, $\mathbf{a}$ is a global minimum for the function $f(\mathbf{x})$ over $\mathcal{L}$. Since the solution of Eq. 2 is in general an NP-hard problem, the exact identification of a global minimum might be very difficult. Therefore, solutions with sufficient good fitness, provided by heuristic procedures, are often considered acceptable for several practical purposes. Among different heuristic procedures, PSO is often the method of choice for its capability to outreach a suitable approximate solution within a few iterations. In PSO, the positions of the particles represent the candidate solutions and will be denoted by $\mathbf{x} \in \mathcal{L}$, with associated fitness $f(\mathbf{x})$. Moreover, in this paper the compact set $\mathcal{L}$ represents the box constraints.

2.1 Original formulation

The original formulation of the PSO algorithm, as presented in [16], reads

$$\begin{cases} \mathbf{v}_i^{k+1} = w\mathbf{v}_i^k + c_1 r_{1,i}^k(\mathbf{x}_{i,pb} - \mathbf{x}_i^k) + c_2 r_{2,i}^k(\mathbf{x}_{gb} - \mathbf{x}_i^k) \\ \mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{cases} \tag{4}$$

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

An overall *constriction factor* χ is used in [34, 35, 36, 37, 38], in place of the inertia weight w. Accordingly, the system in Eq. 4 is recast in the following equivalent form

$$\begin{cases} \mathbf{v}_i^{k+1} = \chi\left[\mathbf{v}_i^k + c_1 r_{1,i}^k(\mathbf{x}_{i,pb} - \mathbf{x}_i^k) + c_2 r_{2,i}^k(\mathbf{x}_{gb} - \mathbf{x}_i^k)\right] \\ \mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{cases} \tag{5}$$

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

$$\chi = \frac{2}{\left|\sqrt{2-\varphi-\sqrt{\varphi^2-4\varphi}}\right|}, \;\; \text{where } \; \varphi = c_1 + c_2, \;\; \varphi > 4 \tag{6}$$

is indicated in [35], where setting the value of φ to 4.1, with $\chi = 0.729$, $c_1 = c_2 = 2.050$ is suggested [36]. Note that PSO schemes including both the parameters w and χ have been also proposed in the literature. Furthermore, recent surveys on PSO coefficients, in the light of possible divergence of particles trajectories, can be found in [39] and [40].

2.2 Deterministic formulation

Two essential requisites in SBD problems are surely represented by the efficiency of the optimization procedure adopted, along with the possibility to replicate the *same procedure* on different settings of the problem. In order to comply with both the latter facts, a deterministic version of the PSO algorithm (namely DPSO) was formulated in [11] by setting $r_{1,i}^k = r_{2,i}^k = 1$ in Eq. 5. Of course, skipping random parameters does not allow to assess a statistical analysis of the results on several practical problems. Nevertheless, since SBD problems are usually computationally very expensive, and represent just one step within a complex and formal project plan, the use of DPSO revealed to be mandatory with respect to adopting PSO. On the overall, the final DPSO scheme becomes

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

In the context of SBD for ship design optimization, as mentioned in Section 1, the formulation of Eq. 7 was compared to the original in [5]. 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 [41], are

$$\begin{cases} 0 < |\chi| < 1 \\ 0 < \omega < 2\,(\chi + 1) \end{cases} \tag{8}$$

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

$$\beta = \frac{\omega}{2(\chi + 1)} \tag{9}$$

and assuming $\chi > 0$ as usually in the literature, the conditions of Eq. 8 reduce to

$$\begin{cases} 0 < \chi < 1 \\ 0 < \beta < 1 \end{cases} \tag{10}$$

2.3 Synchronous and asynchronous implementations

As reported in Section 1, two different implementations of DPSO are comparatively applied in this paper. In particular, the *synchronous* implementation of DPSO (SDPSO) updates the personal bests $\{\mathbf{x}_{i,pb}\}$ and the global best $\mathbf{x}_{gb}$, along with particles velocity and position, at the end of each iteration. SDPSO is presented as a pseudo-code in Alg. 1, and as a block diagram in Fig. 1a. The synchronous update ensures any particle of a perfect and complete information of its neighborhood. However, synchronous update is a costly choice [12], as a particle needs to wait for the whole swarm to be updated, before it can move to a new position and continue its search. Hence, the first particle evaluated might be idle for a long time, waiting for the whole swarm to be updated.

Algorithm 1 SDPSO pseudo-code

```
Initialize a swarm of $N_p$ particles
while ($k <$ Max number of iterations) do
    for $i = 1, N_p$ do
        Evaluate $f(\mathbf{x}_i^k)$
    end for
    Update $\{\mathbf{x}_{i,pb}\}$, $\mathbf{x}_{gb}$
    Update particle positions and velocities $\{\mathbf{x}_i^{k+1}\}$, $\{\mathbf{v}_i^{k+1}\}$
end while
Output the best solution found
```

Furthermore, 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 SDPSO may significantly increase. In contrast to SDPSO, the *asynchronous* implementation of PSO (namely ADPSO [12]) 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. Therefore in this version a particle's position/velocity is updated using partial or "imperfect" information of its neighborhood. I.e., some information come from particles with position/velocity updated in the current iteration, while partial information is provided by particles whose position/velocity was updated in the previous iterations. This last aspect contributes to diversity in the swarm, which is a desirable trait. ADPSO is presented as a pseudo-code in Alg. 2, and as a block diagram in Fig. 1b.

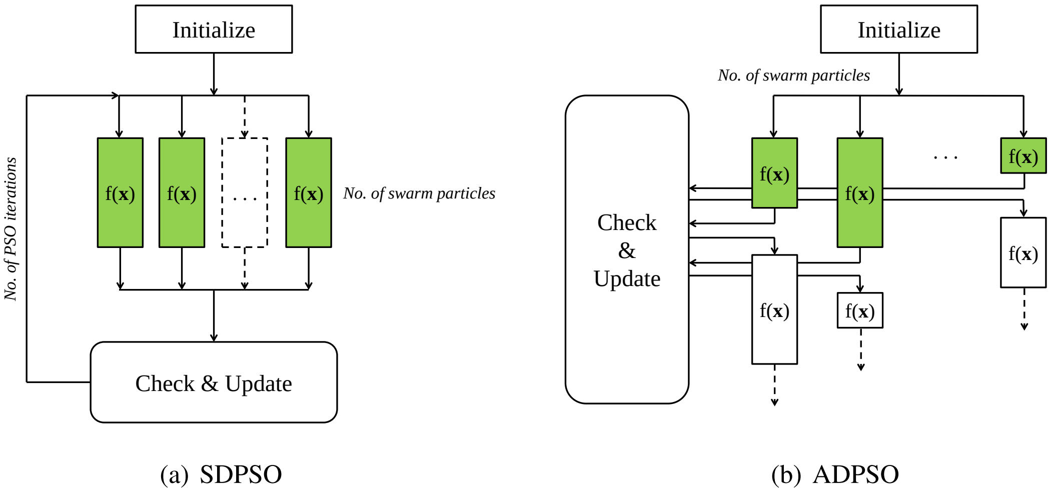
The differences between ADPSO and SDPSO are summarized in Tab. 1.

Algorithm 2 ADPSO pseudo-code

```
Initialize a swarm of $N_p$ particles
while ($k <$ Max number of iterations) do
    for $i = 1, N_p$ do
        evaluate $f(\mathbf{x}_i^k)$
        Update $\{\mathbf{x}_{i,pb}\}$, $\mathbf{x}_{gb}$
        Update particle positions and velocities $\{\mathbf{x}_i^{k+1}\}$, $\{\mathbf{v}_i^{k+1}\}$
    end for
end while
Output the best solution found
```

Table 1: Synchronous and asynchronous DPSO characteristics.

	SDPSO	**ADPSO**
Update	At the end of each iteration	As soon as the objective function is evaluated
Information	The whole swarm uses information from the same iteration	A particle could use information from both current and previous iterations



(a) SDPSO

(b) ADPSO

Figure 1: Block diagrams for parallel SDPSO (a) and ADPSO (b). The green boxes represent the first set of particles evaluated by the algorithm [14].

3 PSO settings and evaluation metrics

The choice of PSO parameters used in the current analysis is defined in the following subsection. Their full-factorial combination is considered, resulting in a total of 420 PSO setups.

3.1 Number of particles

The number of particles used (N_p) is defined as

$$N_p = 2^m N_{dv}\,, \qquad \text{with} \quad m \in \mathbb{N}\,[1,7] \tag{11}$$

therefore ranging from $N_p = 2N_{dv}$ to $N_p = 128N_{dv}$.

3.2 Particles initialization

In order to avoid possible undesired bias due to the choice of the swarm initialization, parameters from the literature are investigated in the current study. In particular, the initialization of particles' location and velocity follows a deterministic and homogeneous distribution, according with the Hammersley sequence sampling (HSS) [30]. Specifically, let $\mathbf{q} = \{q_1, \ldots, q_{N_{dv}-1}\}$ be a vector of prime numbers with $q_i \neq q_j$, $\forall i \neq j$. Any positive integer i can be expressed using the sequence $\{q_j\}$ by

$$i = \sum_{k=0}^{r} a_k q_j^k \tag{12}$$

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

$$\mathbf{x}_i = \left\{ \frac{i}{N_p},\, \phi_{q_1}(i),\, \ldots,\, \phi_{q_{N_{dv}-1}}(i) \right\} \quad \text{for} \;\; i = 0, 1, 2, \ldots, N_p - 1 \tag{13}$$

where $\phi_{q_j}(i) = \sum_{k=0}^{r} a_k / q_j^{k+1}$.

The Eq. 13 is applied to three different sub-domains, defined as:

1. entire domain $\mathcal{L}$ (red dots in Fig. 2a)
2. domain bounds (blu triangles in Fig. 2b)
3. domain and bounds (red dots and blue triangles in Fig. 2c)

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

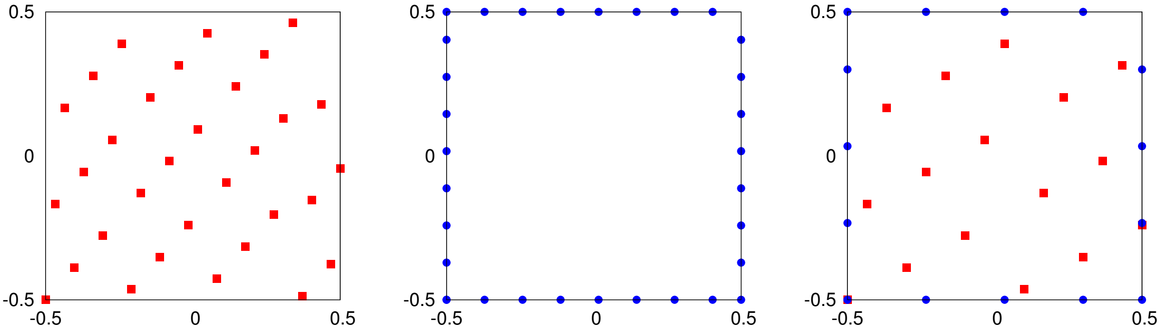
- null velocity:

$$\mathbf{v}_i = 0, \qquad \forall\, i \in [1, N_p] \tag{14}$$

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

$$\mathbf{v}_i = \frac{2}{\sqrt{N_{dv}}}\left(\mathbf{x}_i - \frac{\mathbf{l}+\mathbf{u}}{2}\right) \tag{15}$$

  where $\mathbf{l}$ and $\mathbf{u}$ represent the lower and upper bound for $\mathbf{x}$, respectively [5].



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

Figure 2: Examples of initializations in $\mathcal{L} = [-0.5, 0.5] \times [-0.5, 0.5]$ with 32 particles.

Combining initial position and velocity settings results in six different initializations, summarized in Tab. 2.

Table 2: Swarm 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

3.3 Coefficient set

With a similar approach used in Subsection 3.2, in this paper the authors are first considering a comparison among different setups of DPSO, in order to find a promising configuration, to be applied for the solution of an engineering design problem. On this guideline, on one hand in this paper the authors are not proposing their own set of preferable PSO coefficients. On the other hand, a fair comparison within a panel of different DPSO coefficient set choices requires to test original settings from the literature, without introducing any hybrid combination/integration among them.

To the latter purpose, five coefficient sets are taken from literature, as proposed by several authors. The first set is the original by Shi and Eberhart [16], the second was suggested by Carlisle and Dozier [12], the third was proposed by Trelea [17], the fourth is a further suggestion by Clerc [31], whereas the fifth was suggested by Peri and Tinti [32]. The associated numerical values are included in Tab. 3, and they all satisfy Eq. 10. For possible pros and cons of each coefficient set choice, the reader can refer to the above references.

Table 3: Coefficient set.

ID Set	**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

3.4 Box constraints

The original PSO 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. 7 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 [25, 42, 43, 44]. In particular, the use of exact or sequential penalty functions may be inadequate in the current work, for at least a couple of reasons:

- only box constraints are used (i.e. simple) or inequality constraints, which have been handled in the literature of PSO and other heuristics in a specific way (see Alg. 3 and 4 in this section);
- the use of general penalty approaches described in [25] is expected to be appealing when highly nonlinear constraints are involved. Moreover, a correct implementation of the latter methods definitely requires a fine tuning of the penalty parameters involved, including their iterative update.

Thus, according with the above motivations, and in order to avoid the possible bias introduced by the penalty parameters update (see also [45] for details), herein the approach presented in [22] is applied, since it is specifically designed for the current constrained problems, in the framework of PSO literature.

Specifically, the particles are confined within $\mathcal{L}$ using an inelastic wall-type approach (IW). If a particle is found to violate one of the bounds in the transition from k-th to (k+1)-th PSO iteration, it is placed on the bound setting to zero the associated velocity component (see Fig. 3a). This approach helps the algorithm to explore the domain bounds. The IW approach is implemented in Alg. 3.

Algorithm 3 Inelastic wall-type approach (IW)

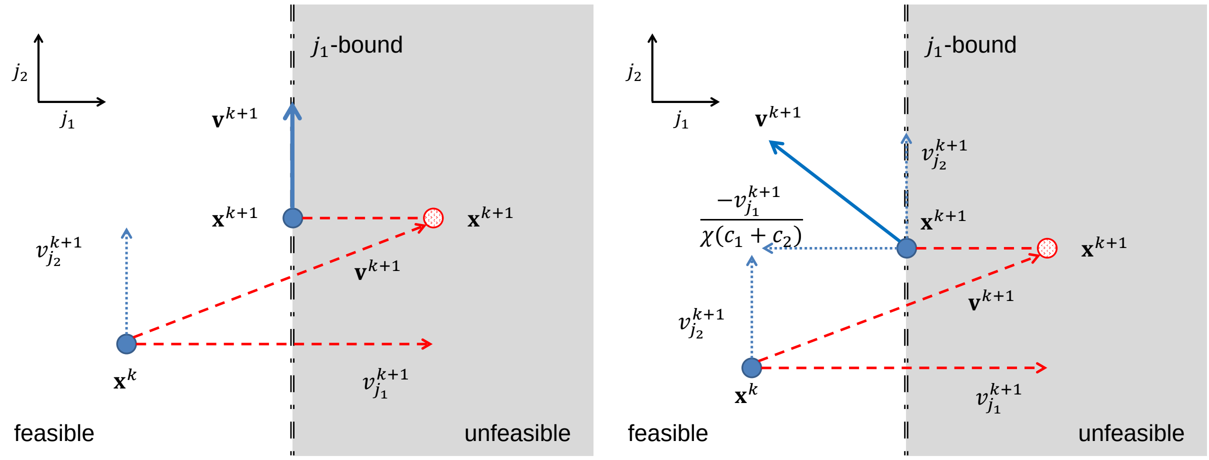
```
for $i = 1$, number of particles do
    for $j = 1$, number of variables do
        if $x^k_{i,j} > u_j$ then
            $x^k_{i,j} = u_j, \quad v^k_{i,j} = 0$
        else {if $x^k_{i,j} < l_j$}
            $x^k_{i,j} = l_j, \quad v^k_{i,j} = 0$
        end if
    end for
end for
```

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 PSO 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. 3b):

Algorithm 4 Semi-elastic wall-type approach (SEW)

```
for $i = 1$, number of particles do
    for $j = 1$, number of variables do
        if $x^k_{i,j} > u_j$ then
            $v^k_{i,j} = -v^k_{i,j}/[\chi(c_1 + c_2)]$
        else {if $x^k_{i,j} < l_j$}
            $v^k_{i,j} = -v^k_{i,j}/[\chi(c_1 + c_2)]$
        end if
    end for
end for
```

Observe that the damping factor $[\chi(c_1 + c_2)]^{-1}$ in Alg. 4 is used to confine the particle in the feasible domain.



(a) Inelastic Wall

(b) Semi-Elastic Wall

Figure 3: Inelastic (a) and semi-elastic (b) wall-type approaches applied in the transition from k-th to $(k + 1)$-th PSO iteration.

3.5 Number of function evaluations and PSO iterations

The maximum number of function evaluations N_{feval} (evaluations budget) in the present work is defined as

$$N_{feval} = 2^n N_{dv}, \quad \text{where} \quad n \in \mathbb{N}\,[7, 12] \tag{16}$$

and therefore ranges from $128N_{dv}$ to $4096N_{dv}$. As a consequence, according with the setting in Eq. 11, the number of PSO iterations N_{iter} performed is given as

$$N_{iter} = \frac{N_{feval}}{N_p} = \frac{2^n N_{dv}}{2^m N_{dv}} = 2^{n-m} \tag{17}$$

3.6 Evaluation metrics

Three absolute performance criteria are used to further assess the algorithms, which are defined as follows [24]:

$$\Delta_x = \sqrt{\frac{1}{N_{dv}} \sum_{j=1}^{N_{dv}} \left(\frac{x_{j,\min} - x^\star_{j,\min}}{R_j} \right)^2}, \quad \Delta_f = \frac{f_{\min} - f^\star_{\min}}{f^\star_{\max} - f^\star_{\min}}, \quad \Delta_t = \sqrt{\frac{\Delta_x^2 + \Delta_f^2}{2}} \tag{18}$$

where Δ_x represents a normalized Euclidean distance between the minimum position found by the algorithm ($\mathbf{x}_{\min}$) and the analytical minimum position ($\mathbf{x}^\star_{\min}$), and $R_j = |u_j - l_j|$ is the range of the j-th variable. Δ_f is the associated normalized distance in the image space, where $f_{\min}$ 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 domain $\mathcal{L}$. Δ_t is a combination of Δ_x and Δ_f and is used for an overall assessment.

Additionally, the relative variability $\sigma^2_{r,k}$ [29] for each metric Δ_x, Δ_f, Δ_t (Eq. 18) is used to assess the impact of each setting parameter s_k on the algorithms' performance. Let $\mathcal{P}$ be a set of $|\mathcal{P}|$ problems, defining the algorithm's setting 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

$$\sigma^2_{r,k} = \frac{\sigma^2_k}{\sum_{k=1}^{|S|} \sigma^2_k} \tag{19}$$

where

$$\sigma^2_k = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left[\hat{\Delta}_k(\omega) \right]^2 - \left[\frac{1}{|\Omega|} \sum_{\omega \in \Omega} \hat{\Delta}_k(\omega) \right]^2 \tag{20}$$

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

$$\hat{\Delta}_k(\omega) = \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \bar{\Delta}(s), \qquad \mathcal{B} = \{s : s_k = \omega\} \tag{21}$$

and

$$\bar{\Delta}(s) = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} [\Delta(s)]_p \tag{22}$$

4 Optimization problems

4.1 Analytical test functions

One hundred analytical test functions are used in the preliminary numerical experience, including a wide variety of problems, such as continuous and discontinuous, differentiable and non-differentiable, separable and non-separable, scalable and non-scalable, unimodal and multimodal, with dimensionality ranging from two to fifty [26, 27, 28, 29, 46]. The numerical experience is conducted setting a part problems with respectively less and more than 10 design variables, as summarized in Tabs. 4 and 5.

Table 4: Analytical test functions with $N_{dv} < 10$.

f_p	**Name**	**Dimension** N_{dv}	**Bounds** $[l, u]^{j,\dots,N_{dv}}$	**Optimum** $f^\star_{\min}$
$f_{1,2}$	5^n loc. minima (Levy)	2, 5	$[-5, 5]^{N_{dv}}$	0.000
$f_{3,4}$	10^n loc. minima (Levy)	2, 5	$[-5, 5]^{N_{dv}}$	0.000
$f_{5,6}$	15^n loc. minima (Levy)	2, 5	$[-5, 5]^{N_{dv}}$	0.000
f_7	Ackley	2	$[-5, 4]^{N_{dv}}$	0.000
$f_{8,9}$	Alpine	2, 5	$[-9, 7]^{N_{dv}}$	0.000
f_{10}	Beale	2	$[-4.5, 4.5]^{N_{dv}}$	0.000
f_{11}	Booth	2	$[-10, 10]^{N_{dv}}$	0.000
f_{12}	Bukin n.6	2	$[-15, -5] \times [-3, 3]$	0.000
f_{13}	Colville	4	$[-10, 10]^{N_{dv}}$	0.000
$f_{14,15}$	Cosine Mixture	2, 4	$[-1, 0.5]^{N_{dv}}$	$-0.100 \cdot N_{dv}$
$f_{16,17}$	Dixon-Price	2, 5	$[-1, 1]^{N_{dv}}$	0.000
f_{18}	Easom	2	$[-100, 100]^{N_{dv}}$	-1.000
$f_{19,20}$	Exponential	2, 4	$[-9, 7]^{N_{dv}}$	-1.000
f_{21}	Freudenstein-Roth	2	$[-5, 5]^{N_{dv}}$	0.000
f_{22}	Goldstein-Price	2	$[-2, 2]^{N_{dv}}$	3.000
$f_{23,24}$	Griewank	2, 5	$[-9, 7]^{N_{dv}}$	0.000
f_{25}	Hartman n.3	3	$[0, 1]^{N_{dv}}$	-3.860
f_{26}	Hartman n.6	6	$[0, 1]^{N_{dv}}$	-3.320
f_{27}	Matyas	2	$[-9, 7]^{N_{dv}}$	0.000
$f_{28,29}$	Multi Modal	2, 5	$[-1, 0.5]^{N_{dv}}$	0.000
f_{30}	Powell	8	$[-4, 5]^{N_{dv}}$	0.000
f_{31}	Quartic	2	$[-10, 10]^{N_{dv}}$	-0.352
f_{32}	Rosenbrock	2	$[-30, 30]^{N_{dv}}$	0.000
f_{33}	Schaffer n.2	2	$[-100, 90]^{N_{dv}}$	0.000
f_{34}	Schaffer n.6	2	$[-100, 90]^{N_{dv}}$	0.000
f_{35}	Schubert penalty 1	2	$[-10, 10]^{N_{dv}}$	-186.731
f_{36}	Schubert penalty 2	2	$[-10, 10]^{N_{dv}}$	-186.731
f_{37}	Shekel n.5	4	$[0, 10]^{N_{dv}}$	-10.153
f_{38}	Shekel n.7	4	$[0, 10]^{N_{dv}}$	-10.403
f_{39}	Shekel n.10	4	$[0, 10]^{N_{dv}}$	-10.536
f_{40}	Six-Hump Camel Back	2	$[-2.5, 2.5] \times [-1.5, 1.5]$	-1.032
f_{41}	Sphere	2	$[-5, 4]^{N_{dv}}$	0.000
$f_{42,43}$	Styblinski-Tang	2, 4	$[-5, 5]^{N_{dv}}$	$-39.166 \cdot N_{dv}$
f_{44}	Test Tube Holder	2	$[-10, 10]^{N_{dv}}$	-10.872
f_{45}	Three-Hump Camel Back	2	$[-5, 4]^{N_{dv}}$	0.000
f_{46}	Treccani	2	$[-5, 4]^{N_{dv}}$	0.000
f_{47}	Tripod	2	$[-100, 100]^{N_{dv}}$	0.000
f_{48}	Vincent	5	$[0.25, 10]^{N_{dv}}$	$-N_{dv}$
f_{49}	Xin-She Yang n.2	5	$[-2\pi, \pi]^{N_{dv}}$	0.000
f_{50}	Xin-She Yang n.4	5	$[-10, 10]^{N_{dv}}$	-1.000

Table 5: Analytical test functions with $N_{dv} \geq 10$.

f_p	**Name**	**Dimension** N_{dv}	**Bounds** $[l,u]^{j,\ldots,N_{dv}}$	**Optimum** $f^\star_{\min}$
$f_{1,2}$	5^n loc. minima (Levy)	10, 20	$[-5,5]^{N_{dv}}$	0.000
$f_{3,4}$	10^n loc. minima (Levy)	10, 20	$[-5,5]^{N_{dv}}$	0.000
$f_{5,6}$	15^n loc. minima (Levy)	10, 20	$[-5,5]^{N_{dv}}$	0.000
$f_{7,8,9}$	Ackley	10, 30, 50	$[-5,4]^{N_{dv}}$	0.000
$f_{10,11}$	Alpine	10, 20	$[-9,7]^{N_{dv}}$	0.000
$f_{12,13,14}$	Dixon-Price	10, 25, 50	$[-1,1]^{N_{dv}}$	0.000
$f_{15,16}$	Griewank	10, 20	$[-9,7]^{N_{dv}}$	0.000
$f_{17,18,19}$	Mishra n.11	10, 25, 50	$[-10,9]^{N_{dv}}$	0.000
$f_{20,21}$	Multi Modal	10, 20	$[-1,0.5]^{N_{dv}}$	0.000
$f_{22,23,24}$	Pathological	10, 25, 50	$[-100,90]^{N_{dv}}$	0.000
f_{25}	Paviani	10	$[2.0001, 9.9999]^{N_{dv}}$	-45.778
$f_{26,27}$	Powell	16, 24	$[-4,5]^{N_{dv}}$	0.000
$f_{28,29,30}$	Rastrigin	10, 30, 50	$[-5.12,5.12]^{N_{dv}}$	0.000
$f_{31,32,33}$	Rosenbrock	10, 25, 50	$[-30,30]^{N_{dv}}$	0.000
$f_{34,35,36}$	Schwefel	10, 25, 50	$[-500,500]^{N_{dv}}$	$-0.100 \cdot N_{dv}$
$f_{37,38,39}$	Trigonometric n.2	10, 25, 50	$[-500,500]^{N_{dv}}$	1.000
$f_{40,41}$	Vincent	15, 30	$[0.25,10]^{N_{dv}}$	$-N_{dv}$
$f_{42,43,44}$	Xin-She Yang n.2	10, 25, 50	$[-2\pi,\pi]^{N_{dv}}$	0.000
$f_{45,46,47}$	Xin-She Yang n.4	10, 25, 50	$[-10,10]^{N_{dv}}$	-1.000
$f_{48,49,50}$	Zacharov	10, 25, 50	$[-5,10]^{N_{dv}}$	0.000

4.2 Hull-form SBD optimization of a high-speed catamaran

The high-speed Delft catamaran [3] is used as SBD test problem. This is an international benchmark geometry used for both experimental and numerical studies, including capabilities of CFD predictions for complex flows with associated validation and global shape optimization for calm water and wave conditions. For the current problem, the objective function $f(\mathbf{x})$ is defined as

$$f(\mathbf{x}) = \frac{R_T}{W} \tag{23}$$

where R_T is the total resistance at Froude number (Fr) equal to 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, beam, and draught. Two feasible design spaces are considered. The first includes overall dimension bounds, whereas the second includes overall dimension bounds and, in addition, constant length between perpendiculars (LBP) [5]. Modifications of the parent hull are performed using high-dimensional free-form deformation (FFD) and 95%-confidence dimensionality reduction based on KLE as shown in [6]. Four variables are used for the first design space and six for the second, referred to in the following as $4D$ and $6D$ space, respectively. New designs g are produced as

$$\mathbf{g}(\mathbf{x}) = \left(1 - \sum_{j=1}^{N_{dv}} x_j\right)\mathbf{g}_0 + \sum_{j=1}^{N_{dv}} x_j\,\mathbf{g}_j \tag{24}$$

where $-1 \leq x_j \leq 1, \forall j \in [1, N_{dv}]$ are the design variables; $N_{dv} = 4$ for the first design space, whereas $N_{dv} = 6$ for the second; g_0 is the original geometry and g_j are the geometries associated to the design space principal directions (or eigenmodes), as provided by KLE for dimensionality reduction. For details, the reader is referred to [5, 6].

Simulations are conducted using the WARP (WAve Resistance Program) code, developed at CNR-INSEAN. Wave resistance computations are based on linear potential flow theory and details of equations, numerical implementation, and validation of the numerical solver are given in [33]. The frictional resistance is estimated using a flat-plate approximation, based on local Reynolds number [47]. 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). Domain bounds are defined by $1LBP$ upstream, $4LBP$ downstream and $2LBP$ sideways.

Since the analytical solution of the optimization problem is not available, the algorithm performance is evaluated considering the objective reduction and its convergence. The design solutions are finally compared with earlier optimization results produced using a high-fidelity CFD solver [5].

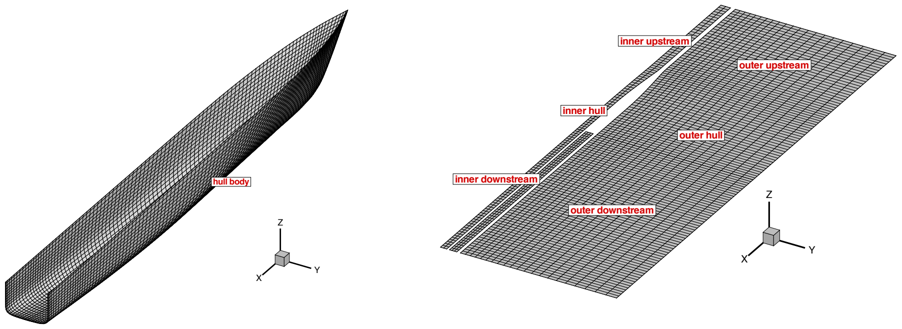


Figure 4: Panel-grid for the CNR-INSEAN WARP potential-flow code.

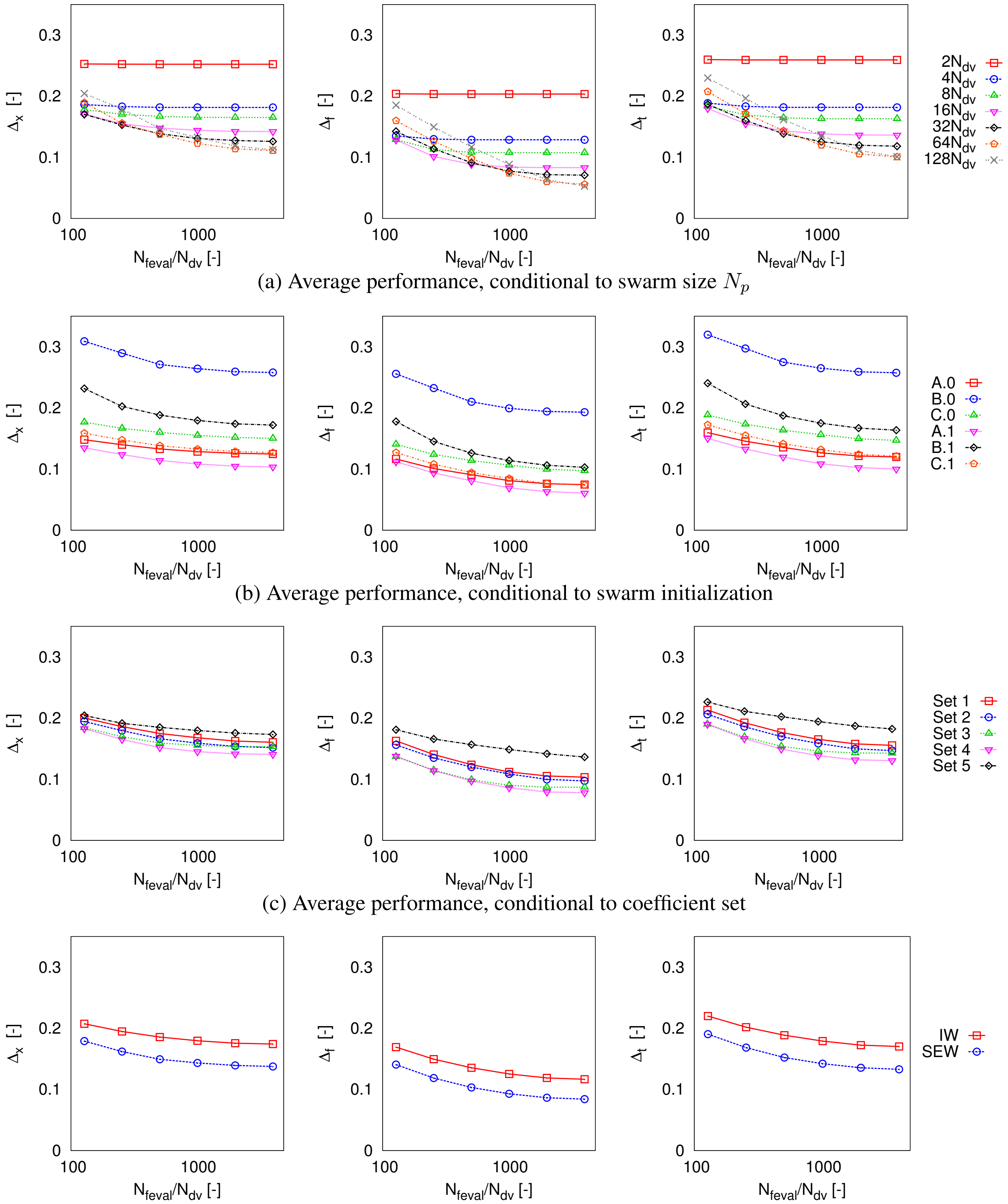
5 Numerical results

5.1 Test functions and DPSO parameters guideline identification

Results on the 100 analytical test functions are presented in the following sections and used to define the guidelines for SDPSO and ADPSO, adopted later for the SBD problem.

5.1.1 SDPSO

Figures 5 and 7, show the performances of SDPSO versus the budget of function evaluations (N_{feval}/N_{dv}), in terms of Δ_x, Δ_f, Δ_t, for $N_{dv} < 10$ and ≥ 10 respectively. Average values are presented, conditional to number of particles, particles initialization, coefficient set, and wall-type approach, respectively. Figures 6 and 8 show the relative variance σ_r^2 of Δ_x, Δ_f, Δ_t for $N_{dv} < 10$ and ≥ 10 respectively, retained by each of the aforementioned parameters. The particles initialization is found the most significant parameter to affect SDPSO performance, especially for $N_{dv} \geq 10$, whereas the coefficient set (selected herein) and the wall-type approach are shown to be the least relevant. Tables 6 and 7 show the five best performing setups for each Δ_x, Δ_f, and Δ_t, for $N_{dv} < 10$ and ≥ 10 respectively, varying the budget of function evaluations. Average values and standard deviations for all SDPSO setups are also provided.



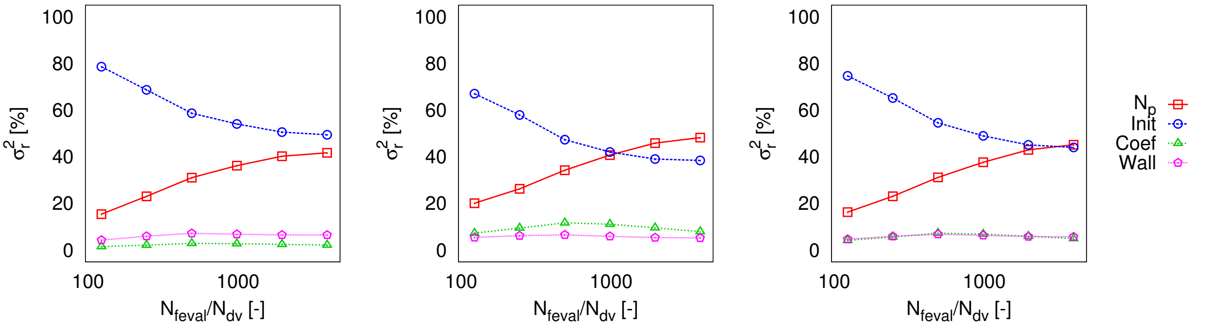
(a) Average performance, conditional to swarm size N_p

(b) Average performance, conditional to swarm initialization

(c) Average performance, conditional to coefficient set

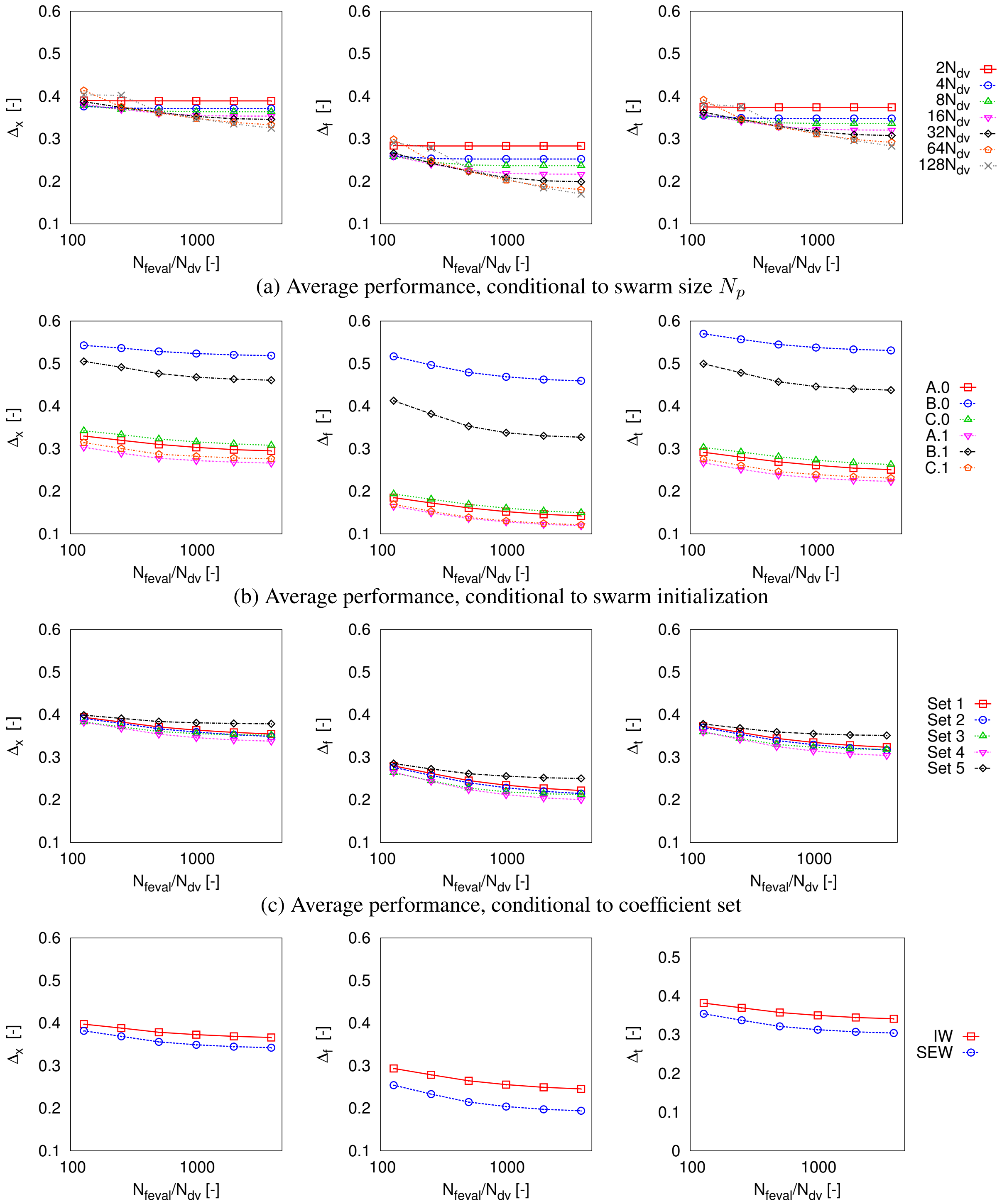
(d) Average performance, conditional to wall-type approach

Figure 5: SDPSO average performance for $N_{dv} < 10$



(a) Δ_x (b) Δ_f (c) Δ_t

Figure 6: Relative variance $\sigma_r^2(\%)$ of SDPSO setting parameters for $N_{dv} < 10$



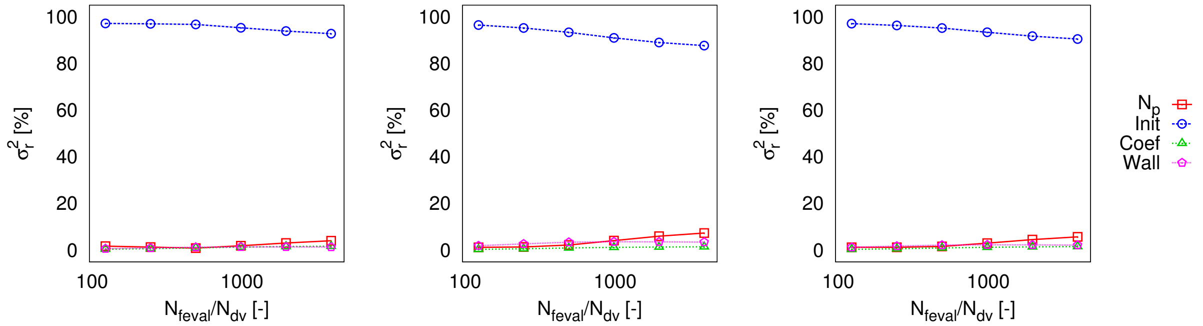
(a) Average performance, conditional to swarm size N_p

(b) Average performance, conditional to swarm initialization

(c) Average performance, conditional to coefficient set

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

Figure 7: SDPSO average performance for $N_{dv} \geq 10$



(a) Δ_x (b) Δ_f (c) Δ_t

Figure 8: Relative variance $\sigma_r^2(\%)$ of SDPSO setting parameters for $N_{dv} \geq 10$

Table 6: Best performing setups for SDPSO, $N_{dv} < 10$.

N_{feval}/N_{dv}	Average (STD)			Best SDPSO														
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_t
				32	A.1	3	SEW	0.099	4	C.1	4	IW	0.033	4	C.1	4	SEW	0.086
	0.193	0.155	0.205	4	C.1	4	SEW	0.102	4	C.0	4	SEW	0.034	4	C.1	4	IW	0.086
128				4	C.1	4	IW	0.106	2	A.1	4	SEW	0.035	2	A.1	4	SEW	0.091
	(0.078)	(0.078)	(0.081)	4	C.1	3	IW	0.106	4	C.1	4	SEW	0.036	4	C.0	4	SEW	0.098
				32	A.1	3	IW	0.107	2	A.1	4	IW	0.046	2	A.1	4	IW	0.100
				32	A.1	3	SEW	0.079	4	C.1	4	SEW	0.031	4	C.1	4	IW	0.082
	0.178	0.134	0.185	64	A.1	3	SEW	0.082	4	C.1	4	IW	0.032	4	C.1	4	SEW	0.082
256				32	A.1	3	IW	0.092	4	C.0	4	SEW	0.032	16	A.1	3	SEW	0.086
	(0.076)	(0.079)	(0.081)	64	A.1	3	IW	0.092	2	A.1	4	SEW	0.035	32	A.1	3	SEW	0.089
				64	A.0	4	SEW	0.097	16	A.1	3	SEW	0.042	2	A.1	4	SEW	0.091
				64	A.1	4	IW	0.070	4	C.1	4	SEW	0.031	32	A.1	3	SEW	0.068
	0.167	0.119	0.170	32	A.1	3	SEW	0.072	4	C.1	4	IW	0.032	64	A.1	3	IW	0.071
512				64	A.1	2	IW	0.072	4	C.0	4	SEW	0.032	64	A.1	3	SEW	0.073
	(0.077)	(0.080)	(0.082)	64	A.1	3	IW	0.074	2	A.1	4	SEW	0.035	64	A.1	4	IW	0.078
				64	A.1	3	SEW	0.076	32	A.1	3	SEW	0.036	4	C.1	4	IW	0.081
				64	A.1	4	IW	0.063	64	A.1	3	SEW	0.021	64	A.1	3	SEW	0.052
	0.161	0.109	0.161	64	A.1	3	SEW	0.063	32	A.1	3	SEW	0.021	64	A.1	4	IW	0.054
1024				64	A.1	2	IW	0.065	64	A.0	4	IW	0.023	32	A.1	3	SEW	0.056
	(0.078)	(0.083)	(0.085)	64	A.1	2	SEW	0.066	64	A.1	3	IW	0.023	64	A.1	3	IW	0.057
				32	A.1	3	SEW	0.070	32	A.0	4	SEW	0.023	64	A.1	4	SEW	0.062
				64	A.1	2	IW	0.053	64	A.1	3	SEW	0.020	64	A.1	2	IW	0.045
	0.157	0.103	0.154	64	A.1	4	IW	0.059	64	A.1	4	SEW	0.020	64	A.1	4	IW	0.049
2048				64	A.1	3	SEW	0.061	64	A.1	4	IW	0.020	64	A.1	3	SEW	0.050
	(0.080)	(0.085)	(0.088)	64	A.1	2	SEW	0.063	64	A.0	4	SEW	0.020	64	A.1	2	SEW	0.052
				64	A.1	3	IW	0.067	32	A.0	4	SEW	0.021	64	A.1	3	IW	0.055
				64	A.1	2	IW	0.049	128	A.1	3	IW	0.020	64	A.1	2	IW	0.042
	0.156	0.100	0.152	64	A.1	4	IW	0.058	128	C.1	3	SEW	0.020	64	A.1	4	IW	0.048
4096				64	A.1	2	SEW	0.059	64	A.1	2	IW	0.020	64	A.1	2	SEW	0.049
	(0.081)	(0.086)	(0.089)	64	A.1	3	SEW	0.060	128	C.1	4	SEW	0.020	64	A.1	3	SEW	0.050
				128	C.1	4	IW	0.064	64	C.0	4	SEW	0.020	128	C.1	4	IW	0.053
				32	A.1	3	SEW	0.076	4	C.1	4	SEW	0.032	4	C.1	4	SEW	0.081
	0.169	0.120	0.171	64	A.1	3	SEW	0.078	4	C.1	4	IW	0.032	4	C.1	4	IW	0.081
Av.				64	A.1	2	IW	0.078	4	C.0	4	SEW	0.033	32	A.1	3	SEW	0.082
	(0.077)	(0.079)	(0.082)	64	A.1	4	IW	0.079	2	A.1	4	SEW	0.035	64	A.1	3	SEW	0.082
				64	A.1	3	IW	0.082	32	A.1	3	SEW	0.040	64	A.1	3	IW	0.083

Table 7: Best performing setups for SDPSO, $N_{dv} \geq 10$.

N_{feval}/N_{dv}	Average (STD)			Best SDPSO														
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_t
				4	A.1	4	SEW	0.261	4	A.1	4	SEW	0.118	4	A.1	4	SEW	0.221
	0.390	0.274	0.368	2	A.1	4	SEW	0.266	8	A.1	3	SEW	0.121	2	A.1	2	IW	0.224
128				8	A.1	3	SEW	0.267	4	A.1	3	SEW	0.122	2	A.1	4	SEW	0.225
	(0.101)	(0.150)	(0.128)	4	A.1	3	SEW	0.267	2	A.1	4	SEW	0.122	8	A.1	3	SEW	0.227
				8	C.1	3	SEW	0.268	4	C.1	4	SEW	0.122	4	A.1	3	SEW	0.227
				4	A.1	4	SEW	0.257	8	A.1	3	SEW	0.107	4	A.1	4	SEW	0.214
	0.379	0.256	0.354	8	A.1	3	SEW	0.260	4	A.1	4	SEW	0.111	8	A.1	3	SEW	0.216
256				16	A.1	3	SEW	0.260	4	C.1	4	SEW	0.113	8	C.1	4	SEW	0.219
	(0.102)	(0.148)	(0.128)	8	C.1	4	SEW	0.261	8	C.1	4	SEW	0.114	8	C.1	3	SEW	0.220
				8	C.1	3	SEW	0.261	8	C.1	3	SEW	0.115	2	A.1	2	IW	0.220
				32	A.1	3	SEW	0.250	16	A.1	3	SEW	0.100	16	A.1	3	SEW	0.208
	0.367	0.240	0.340	16	A.1	3	SEW	0.253	8	A.1	3	SEW	0.104	32	A.1	3	SEW	0.212
512				16	A.1	4	SEW	0.253	8	C.1	4	SEW	0.106	8	C.1	4	SEW	0.212
	(0.103)	(0.147)	(0.128)	8	C.1	4	SEW	0.255	16	C.1	3	SEW	0.107	8	A.1	3	SEW	0.213
				64	A.1	3	SEW	0.255	32	A.1	3	SEW	0.109	4	A.1	4	SEW	0.213
				64	A.1	3	SEW	0.242	32	A.1	3	SEW	0.095	32	A.1	3	SEW	0.200
	0.361	0.230	0.332	32	A.1	3	SEW	0.244	16	A.1	3	SEW	0.097	64	A.1	3	SEW	0.204
1024				16	A.1	4	SEW	0.246	32	C.1	3	SEW	0.099	16	A.1	4	SEW	0.205
	(0.104)	(0.148)	(0.129)	32	A.1	4	SEW	0.247	16	C.1	4	SEW	0.101	16	A.1	3	SEW	0.205
				16	A.1	3	SEW	0.250	64	A.1	3	SEW	0.103	32	A.1	4	SEW	0.207
				64	A.1	3	SEW	0.235	32	A.1	3	SEW	0.092	64	A.1	3	SEW	0.194
	0.357	0.223	0.326	32	A.1	4	SEW	0.240	64	A.1	3	SEW	0.092	32	A.1	3	SEW	0.198
2048				32	A.1	3	SEW	0.242	32	C.1	3	SEW	0.096	32	A.1	4	SEW	0.199
	(0.105)	(0.149)	(0.130)	64	A.1	4	SEW	0.245	16	A.1	3	SEW	0.097	16	A.1	4	SEW	0.203
				16	A.1	4	SEW	0.245	64	C.1	3	SEW	0.097	64	A.1	4	SEW	0.204
				64	A.1	3	SEW	0.233	64	A.1	3	SEW	0.090	64	A.1	3	SEW	0.191
	0.354	0.220	0.323	128	A.1	3	SEW	0.239	32	A.1	3	SEW	0.092	128	A.1	3	SEW	0.196
4096				128	A.1	4	SEW	0.239	128	C.1	3	SEW	0.094	64	A.1	4	SEW	0.197
	(0.105)	(0.150)	(0.131)	64	A.1	4	SEW	0.239	64	C.1	4	SEW	0.094	128	A.1	4	SEW	0.197
				32	A.1	4	SEW	0.239	64	C.1	3	SEW	0.094	64	A.1	4	IW	0.197
				32	A.1	3	SEW	0.254	8	A.1	3	SEW	0.107	16	A.1	3	SEW	0.214
	0.368	0.240	0.340	16	A.1	3	SEW	0.256	16	A.1	3	SEW	0.108	4	A.1	4	SEW	0.215
Av.				4	A.1	4	SEW	0.257	4	A.1	4	SEW	0.111	32	A.1	3	SEW	0.216
	(0.102)	(0.147)	(0.128)	16	A.1	4	SEW	0.257	8	C.1	4	SEW	0.112	8	A.1	3	SEW	0.216
				8	C.1	4	SEW	0.259	32	A.1	3	SEW	0.114	8	C.1	4	SEW	0.217

5.1.2 ADPSO

Generally, ADPSO results are found similar to SDPSO. Specifically, Figs. 9 and 11, show the performances of ADPSO versus the budget of function evaluations, in terms of Δ_x, Δ_f, Δ_t, for $N_{dv} < 10$ and ≥ 10 respectively. Average values are presented, conditional to number of particles, particles initialization, coefficient set, and wall-type approach respectively. Figures 10 and 12 show the relative variance σ_r^2 of Δ_x, Δ_f, Δ_t for $N_{dv} < 10$ and ≥ 10 respectively, retained by each of the aforementioned parameters. The particles initialization is again the most significant parameter to affect ADPSO performance, especially for low budgets and $N_{dv} \geq 10$. 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 9 and 10 summarize the five best performing setups for each Δ_x, Δ_f, and Δ_t, for $N_{dv} < 10$ and ≥ 10, respectively, varying the budget of function evaluations. Overall averages and standard deviations for all ADPSO setups are also included.

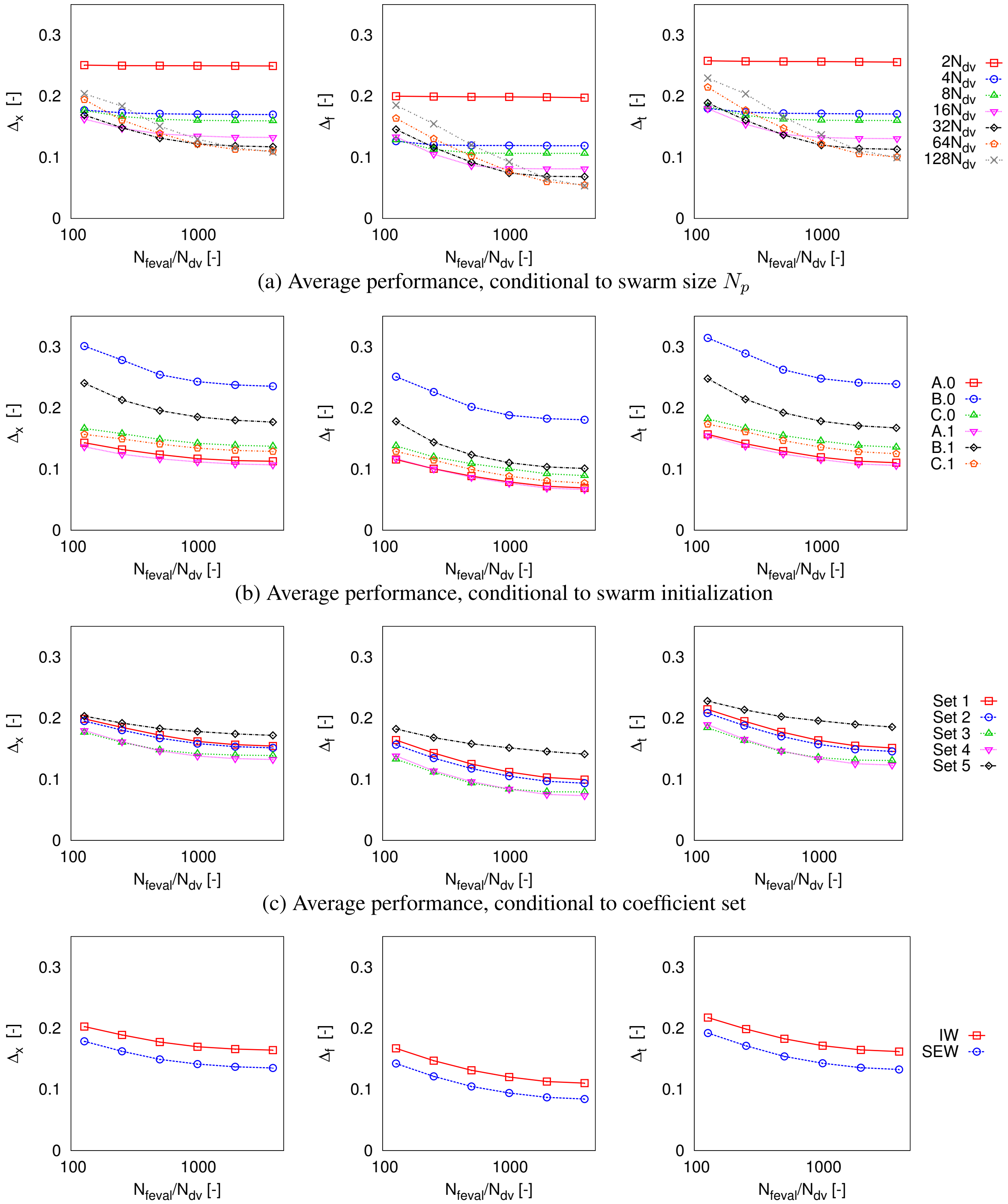
5.1.3 Suggested guidelines

The most promising setups are selected from Tabs. 6, 7, 9, and 10, in order to define a reasonable and robust guideline for the use of SDPSO and ADPSO. These are summarized in Tab. 8.

Figures 13 and 14 show the performance of the suggested setups, for SDPSO and ADPSO and $N_{dv} < 10$ 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 ADPSO is always equivalent or slightly better than SDPSO.

Table 8: Suggested guideline for SDPSO and ADPSO.

	N_{dv}	N_p/N_{dv}	Initialization	coefficient set	Wall-type
SDPSO	< 10	4	C.1	4	IW/SEW
	≥ 10	4	A.1	4	SEW
ADPSO	< 10	4	C.1	4	IW/SEW
	≥ 10	4	A.1	4	SEW



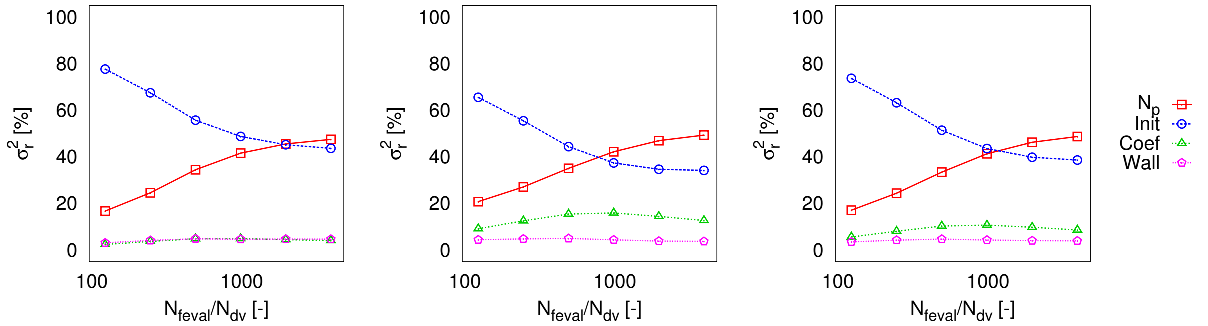
(a) Average performance, conditional to swarm size N_p

(b) Average performance, conditional to swarm initialization

(c) Average performance, conditional to coefficient set

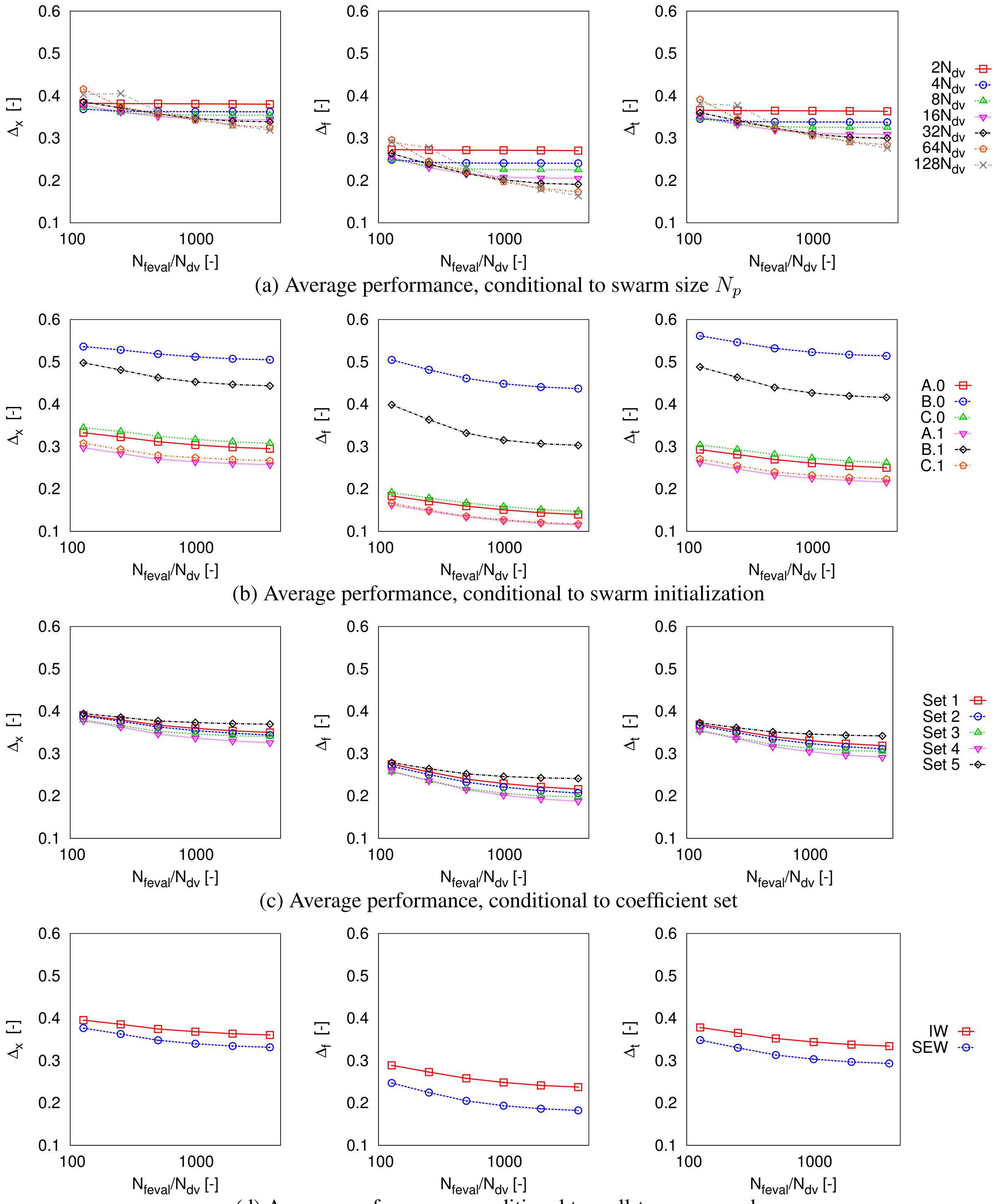
(d) Average performance, conditional to wall-type approach

Figure 9: ADPSO average performance for $N_{dv} < 10$



(a) Δ_x (b) Δ_f (c) Δ_t

Figure 10: Relative variance $\sigma_r^2(\%)$ of ADPSO setting parameters for $N_{dv} < 10$



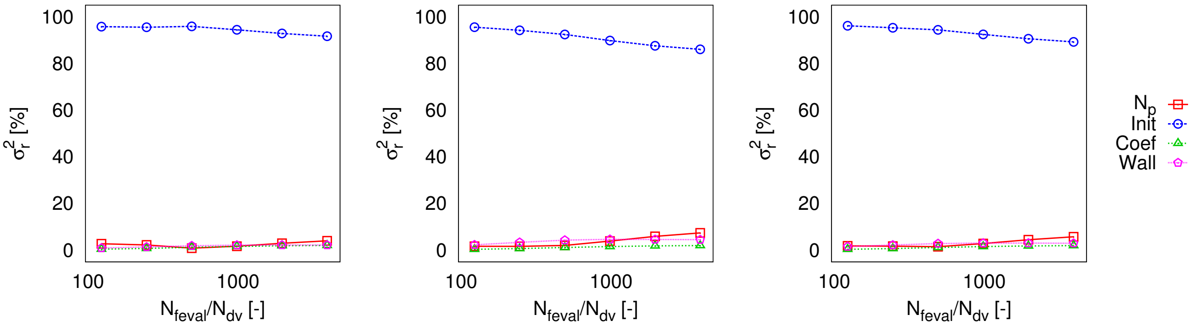
(a) Average performance, conditional to swarm size N_p

(b) Average performance, conditional to swarm initialization

(c) Average performance, conditional to coefficient set

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

Figure 11: ADPSO average performance for $N_{dv} \geq 10$



(a) Δ_x (b) Δ_f (c) Δ_t

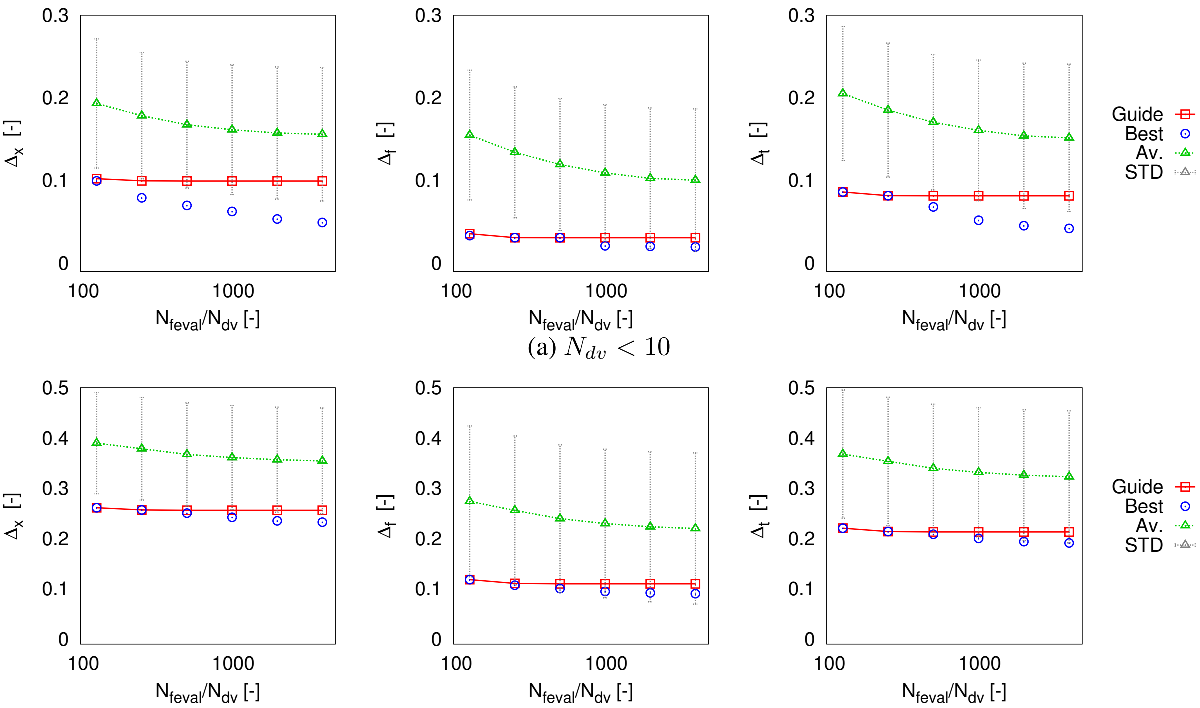
Figure 12: Relative variance $\sigma_r^2(\%)$ of ADPSO setting parameters for $N_{dv} \geq 10$

Table 9: Best performing setups for ADPSO, $N_{dv} < 10$.

N_{feval}/N_{dv}	Average (STD)			Best ADPSO														
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_t
128	0.191 (0.078)	0.155 (0.076)	0.205 (0.080)	4	C.1	4	IW	0.095	4	C.1	3	IW	0.023	4	C.1	4	IW	0.075
				4	C.1	4	SEW	0.097	4	C.1	4	IW	0.024	4	C.1	4	SEW	0.084
				8	A.1	3	IW	0.100	4	C.0	3	IW	0.034	4	C.1	3	IW	0.087
				8	A.1	3	SEW	0.107	4	C.0	4	SEW	0.039	4	C.0	3	IW	0.094
				2	A.1	4	SEW	0.110	4	C.0	4	IW	0.039	4	C.0	4	SEW	0.096
256	0.176 (0.075)	0.134 (0.076)	0.185 (0.079)	4	C.1	4	IW	0.085	4	C.1	4	IW	0.021	4	C.1	4	IW	0.068
				32	A.1	3	SEW	0.092	4	C.1	3	IW	0.023	4	C.1	4	SEW	0.080
				4	C.1	4	SEW	0.093	4	C.0	4	IW	0.031	4	C.1	3	IW	0.085
				8	A.1	3	IW	0.096	4	C.0	3	IW	0.033	4	C.0	4	SEW	0.087
				32	A.1	4	IW	0.097	4	C.0	4	SEW	0.034	4	C.0	3	IW	0.090
512	0.163 (0.073)	0.118 (0.077)	0.168 (0.078)	32	A.1	4	IW	0.073	4	C.1	4	IW	0.020	4	C.1	4	IW	0.065
				64	A.1	3	SEW	0.080	4	C.1	3	IW	0.023	16	A.0	4	IW	0.074
				4	C.1	4	IW	0.082	16	A.0	4	IW	0.024	4	C.1	4	SEW	0.078
				32	C.1	4	IW	0.084	4	C.0	4	IW	0.031	32	A.1	3	SEW	0.079
				32	A.1	3	SEW	0.084	4	C.0	3	IW	0.033	64	A.1	3	SEW	0.082
1024	0.156 (0.075)	0.107 (0.079)	0.157 (0.081)	32	A.1	4	IW	0.064	4	C.1	4	IW	0.020	32	A.1	4	IW	0.057
				64	A.1	3	SEW	0.071	16	A.0	4	IW	0.021	64	A.1	3	SEW	0.059
				64	A.1	4	SEW	0.072	4	C.1	3	IW	0.023	4	C.1	4	IW	0.065
				32	A.0	4	SEW	0.077	64	A.1	3	SEW	0.024	64	A.0	3	SEW	0.067
				64	A.0	4	IW	0.077	64	A.0	3	SEW	0.024	64	A.0	4	SEW	0.067
2048	0.151 (0.076)	0.100 (0.082)	0.150 (0.083)	64	A.1	4	SEW	0.064	4	C.1	4	IW	0.020	64	A.1	4	SEW	0.053
				32	A.1	4	IW	0.064	32	C.1	4	SEW	0.021	32	A.1	4	IW	0.056
				64	A.1	3	SEW	0.070	16	A.0	4	IW	0.021	64	A.1	3	SEW	0.057
				64	A.1	1	IW	0.073	64	A.0	4	IW	0.022	64	A.0	4	IW	0.059
				64	A.0	4	SEW	0.074	32	A.1	2	IW	0.022	64	A.0	4	SEW	0.060
4096	0.150 (0.077)	0.097 (0.083)	0.147 (0.085)	64	A.1	4	SEW	0.063	128	A.1	4	SEW	0.020	64	A.1	4	SEW	0.052
				32	A.1	4	IW	0.064	4	C.1	4	IW	0.020	32	A.1	4	IW	0.056
				64	A.1	3	SEW	0.069	32	A.1	2	IW	0.021	64	A.1	3	SEW	0.056
				64	A.1	1	IW	0.070	32	C.1	4	SEW	0.021	64	A.1	1	IW	0.057
				128	A.0	2	SEW	0.072	16	A.0	4	IW	0.021	64	A.0	4	IW	0.059
Av.	0.164 (0.073)	0.119 (0.076)	0.169 (0.078)	32	A.1	4	IW	0.079	4	C.1	4	IW	0.021	4	C.1	4	IW	0.067
				4	C.1	4	IW	0.084	4	C.1	3	IW	0.023	4	C.1	4	SEW	0.080
				64	A.1	3	SEW	0.089	4	C.0	4	IW	0.033	32	A.1	4	IW	0.085
				32	A.1	3	SEW	0.090	4	C.0	3	IW	0.033	4	C.1	3	IW	0.085
				32	C.1	4	IW	0.090	4	C.0	4	SEW	0.035	32	A.1	3	SEW	0.088

Table 10: Best performing setups for ADPSO, $N_{dv} \geq 10$.

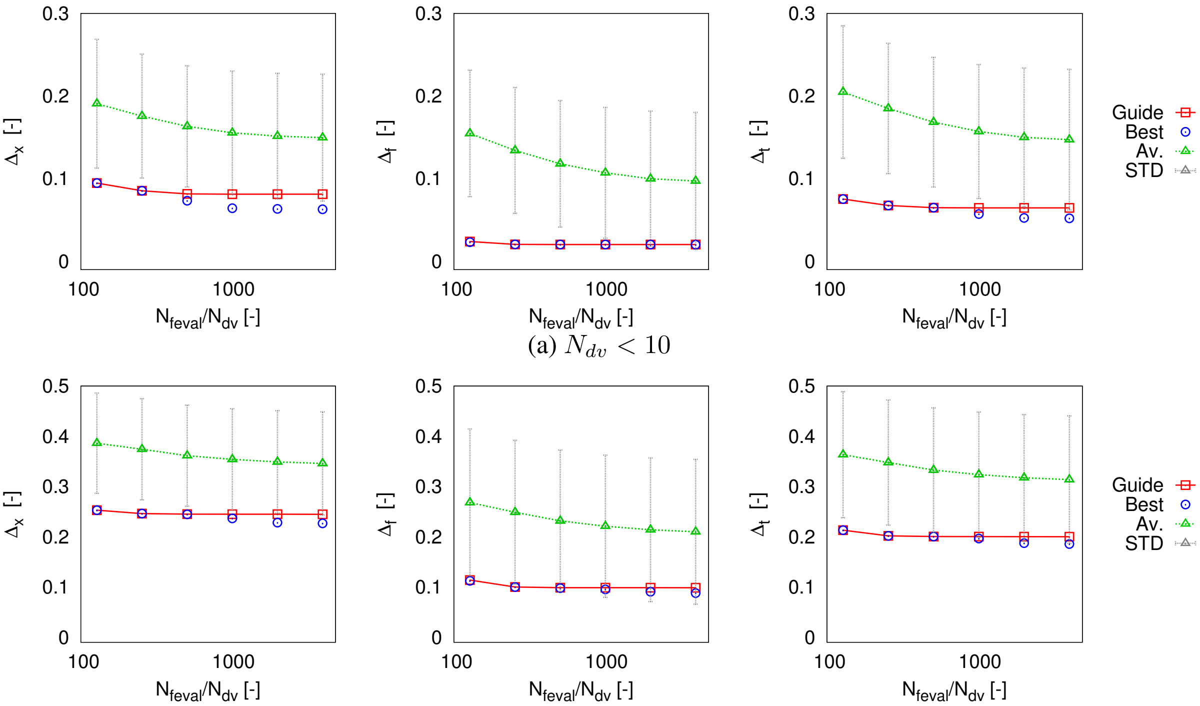
N_{feval}/N_{dv}	Average (STD)			Best ADPSO														
	Δ_x	Δ_f	Δ_t	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_x	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_f	$\frac{N_p}{N_{dv}}$	Init	Coef	Wall	Δ_t
128	0.386 (0.100)	0.268 (0.146)	0.363 (0.125)	4	A.1	4	SEW	0.253	4	A.1	3	SEW	0.112	4	A.1	4	SEW	0.213
				2	A.1	4	SEW	0.255	2	C.1	4	SEW	0.113	2	A.1	4	SEW	0.216
				4	C.1	4	SEW	0.256	4	A.1	4	SEW	0.114	4	C.1	4	SEW	0.217
				4	C.1	2	SEW	0.257	2	A.1	1	IW	0.115	2	A.1	2	IW	0.218
				2	A.1	2	SEW	0.260	4	C.1	4	SEW	0.117	4	A.1	3	SEW	0.220
256	0.374 (0.101)	0.249 (0.143)	0.349 (0.125)	4	A.1	4	SEW	0.246	4	A.1	4	SEW	0.100	4	A.1	4	SEW	0.202
				4	C.1	4	SEW	0.250	8	A.1	3	SEW	0.108	4	C.1	4	SEW	0.209
				4	C.1	2	SEW	0.252	4	C.1	4	SEW	0.109	8	C.1	4	SEW	0.212
				8	C.1	4	SEW	0.252	4	A.1	3	SEW	0.110	4	C.1	2	IW	0.213
				2	A.1	4	SEW	0.254	8	C.1	3	SEW	0.110	2	A.1	4	SEW	0.215
512	0.361 (0.101)	0.232 (0.141)	0.333 (0.124)	8	C.1	4	SEW	0.245	8	C.1	4	SEW	0.098	4	A.1	4	SEW	0.200
				4	A.1	4	SEW	0.245	4	A.1	4	SEW	0.099	8	C.1	4	SEW	0.200
				16	A.1	4	SEW	0.246	16	A.1	3	SEW	0.103	16	A.1	4	SEW	0.206
				4	C.1	4	SEW	0.249	8	A.1	3	SEW	0.104	8	A.1	4	SEW	0.206
				16	C.1	4	SEW	0.250	8	A.1	4	SEW	0.104	4	C.1	4	SEW	0.207
1024	0.354 (0.101)	0.221 (0.142)	0.324 (0.125)	32	A.1	4	SEW	0.237	8	C.1	4	SEW	0.095	16	A.1	4	SEW	0.197
				16	A.1	4	SEW	0.239	16	A.1	4	SEW	0.097	8	C.1	4	SEW	0.198
				16	C.1	4	SEW	0.242	16	C.1	4	SEW	0.097	16	C.1	4	SEW	0.199
				32	A.1	4	IW	0.243	4	A.1	4	SEW	0.099	32	A.1	4	SEW	0.200
				8	C.1	4	SEW	0.243	16	A.1	3	SEW	0.099	4	A.1	4	SEW	0.200
2048	0.349 (0.102)	0.214 (0.143)	0.317 (0.126)	32	A.1	4	SEW	0.228	32	A.1	4	SEW	0.091	32	A.1	4	SEW	0.187
				64	A.1	3	SEW	0.233	32	C.1	4	SEW	0.093	64	A.1	3	SEW	0.194
				16	A.1	4	SEW	0.238	64	A.1	3	SEW	0.095	16	A.1	4	SEW	0.195
				32	A.1	4	IW	0.239	16	C.1	4	SEW	0.095	32	A.1	4	IW	0.196
				16	C.1	4	SEW	0.239	16	A.1	4	SEW	0.095	16	C.1	4	SEW	0.196
4096	0.346 (0.103)	0.210 (0.144)	0.314 (0.127)	32	A.1	4	SEW	0.227	64	C.1	4	SEW	0.088	32	A.1	4	SEW	0.186
				64	A.1	3	SEW	0.231	32	A.1	4	SEW	0.089	64	A.1	3	SEW	0.190
				128	A.1	3	SEW	0.236	64	A.1	3	SEW	0.092	64	A.1	4	IW	0.193
				64	A.1	4	IW	0.236	32	C.1	4	SEW	0.092	128	A.1	3	SEW	0.193
				64	A.1	4	SEW	0.237	64	A.1	4	SEW	0.092	64	A.1	4	SEW	0.193
Av.	0.362 (0.100)	0.232 (0.141)	0.333 (0.124)	4	A.1	4	SEW	0.246	4	A.1	4	SEW	0.101	4	A.1	4	SEW	0.203
				8	C.1	4	SEW	0.248	8	C.1	4	SEW	0.104	8	C.1	4	SEW	0.206
				16	A.1	4	SEW	0.249	8	A.1	3	SEW	0.108	4	C.1	4	SEW	0.209
				32	A.1	4	SEW	0.250	4	C.1	4	SEW	0.109	16	A.1	4	SEW	0.211
				4	C.1	4	SEW	0.250	8	A.1	4	SEW	0.109	8	A.1	4	SEW	0.211



(a) $N_{dv} < 10$

(b) $N_{dv} \geq 10$

Figure 13: Performance of suggested guidelines using SDPSO



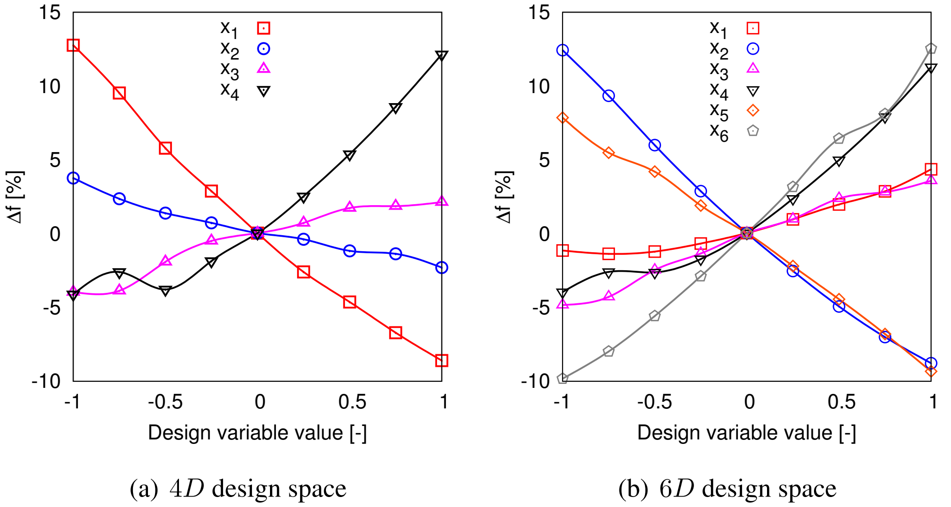
(a) $N_{dv} < 10$

(b) $N_{dv} \geq 10$

Figure 14: Performance of suggested guidelines using ADPSO

5.2 High-speed catamaran SBD optimization

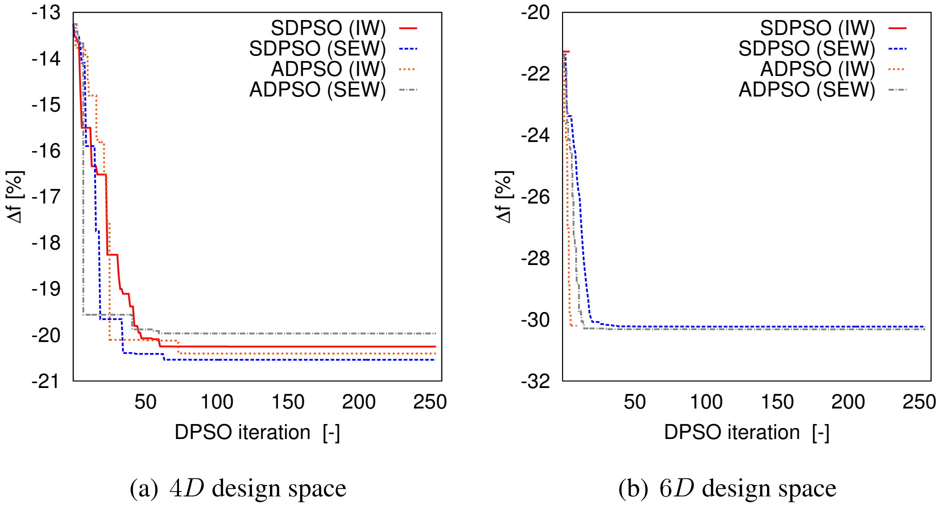
A preliminary sensitivity analysis for each design variable is shown in Fig. 15, showing $\Delta f(\%)$ compared to the parent hull. Changes in f reveal a reduction of the objective function close to 9% for the $4D$ design space and close to 10% for the $6D$. For the current study, all design variables are deemed required for the hull form optimization, each one providing a significant variation of the design objective. Accordingly, the sensitivity analysis is not applied for dimensionality reduction.



(a) $4D$ design space (b) $6D$ design space

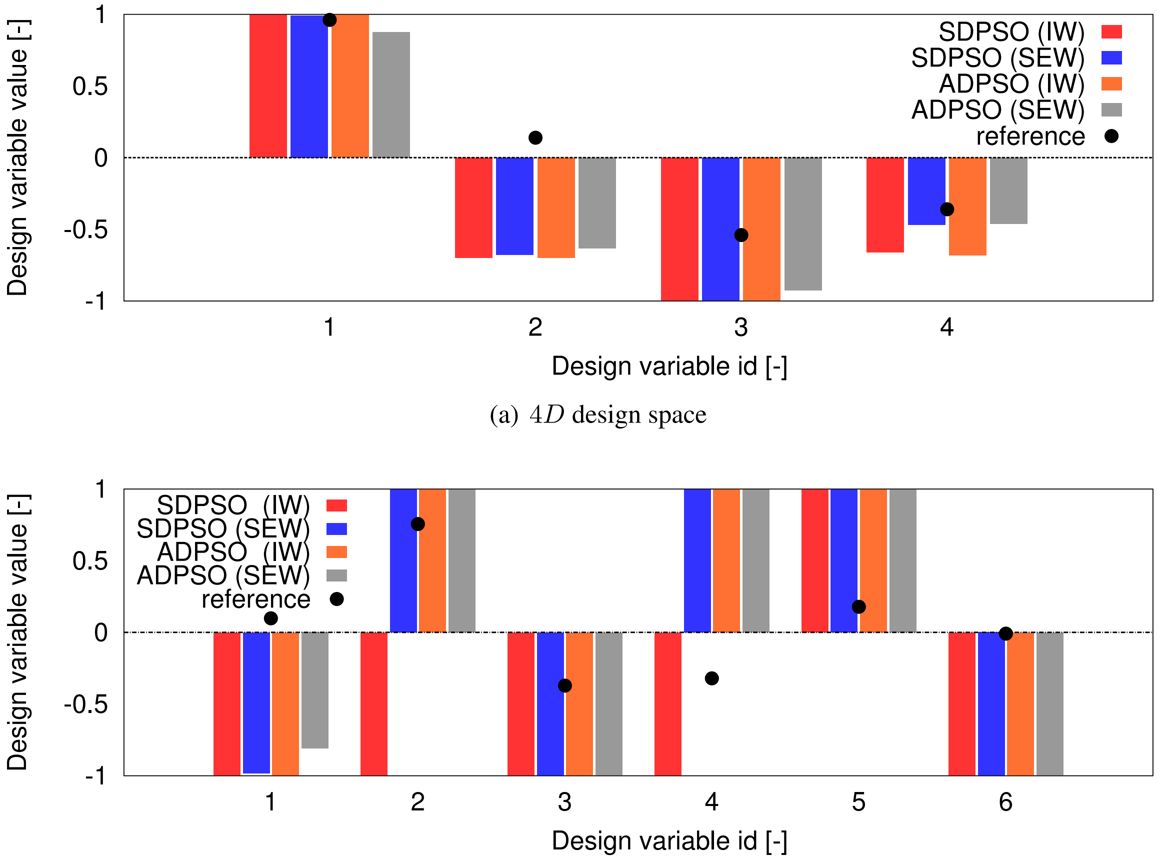
Figure 15: Sensitivity analysis.

The optimization is performed with both SDPSO and ADPSO, as per the guideline suggested in Tab. 8. SDPSO and ADPSO iterations are shown in Fig. 16, revealing a quite fast convergence. It may be noted that the final configuration for the first design space ($4D$) is fairly close (except for the second variable) to that obtained using metamodels and a URANS solver in [5], used here for reference (Fig. 17a). For the second design space ($6D$) the differences are more significant (Fig. 17b). As a general result, one can also observe by the optimization results (see Tab. 11) that SDPSO and ADPSO with both IW and SEW lead to a reduction of the objective function close to 20%, for the $4D$ design space, and greater than 20% for the $6D$ design space.



(a) $4D$ design space (b) $6D$ design space

Figure 16: Convergence of SDPSO and ADPSO.



(a) $4D$ design space

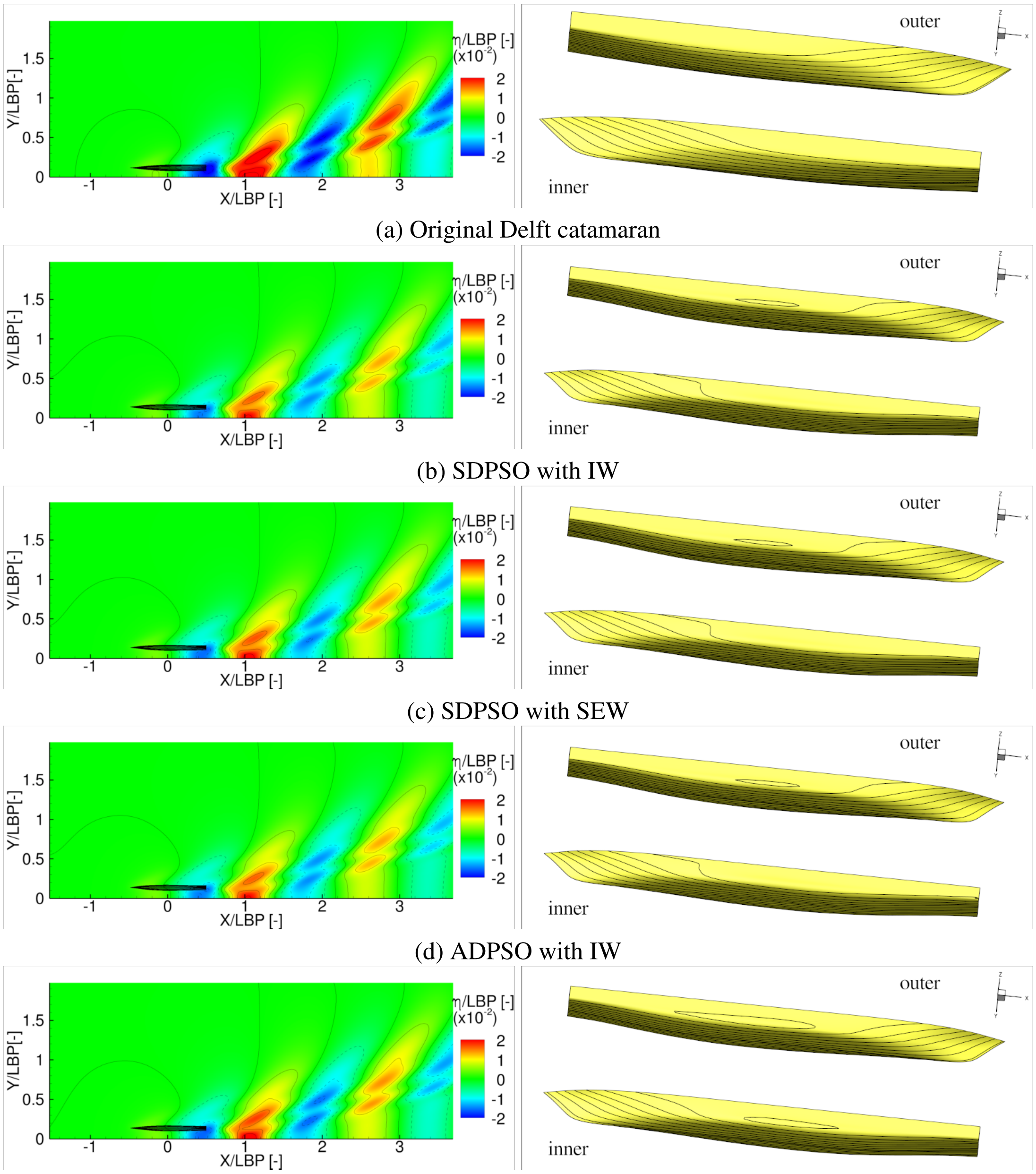
(b) $6D$ design space

Figure 17: Comparison between optimal design variables of SDPSO, ADPSO with IW and SEW by PF and those obtained by metamodels with URANS in [5].

Furthermore, the optimum configuration leads to a considerable reduction of wave's elevation compared to the original shape (Figs. 18 and 19). There are not significant differences between the results obtained by SDPSO and ADPSO, except for SDPSO with IW for the $6D$ design space. In this case the IW approach induces the optimization to stop after 6 iterations. As shown in Fig. 17, differences in optimal design variables are mainly due to IW or SEW.

Table 11: SBD results.

		Wall-type	R_T[N]	W[N]	$f[-]$	$\Delta f[\%]$
Design space	Original		50.15	852.5	5.88e-2	–
$4D$	SDPSO	IW	39.92	850.9	4.69e-2	-20.24
		SEW	40.36	850.6	4.67e-2	-20.57
	ADPSO	IW	39.85	851.1	4.68e-2	-20.41
		SEW	40.68	851.7	4.71e-2	-19.90
$6D$	SDPSO	IW	39.41	849.1	4.64e-2	-21.10
		SEW	34.29	835.4	4.10e-2	-30.27
	ADPSO	IW	34.31	835.5	4.11e-2	-30.10
		SEW	34.08	830.8	4.09e-2	-30.30



(a) Original Delft catamaran

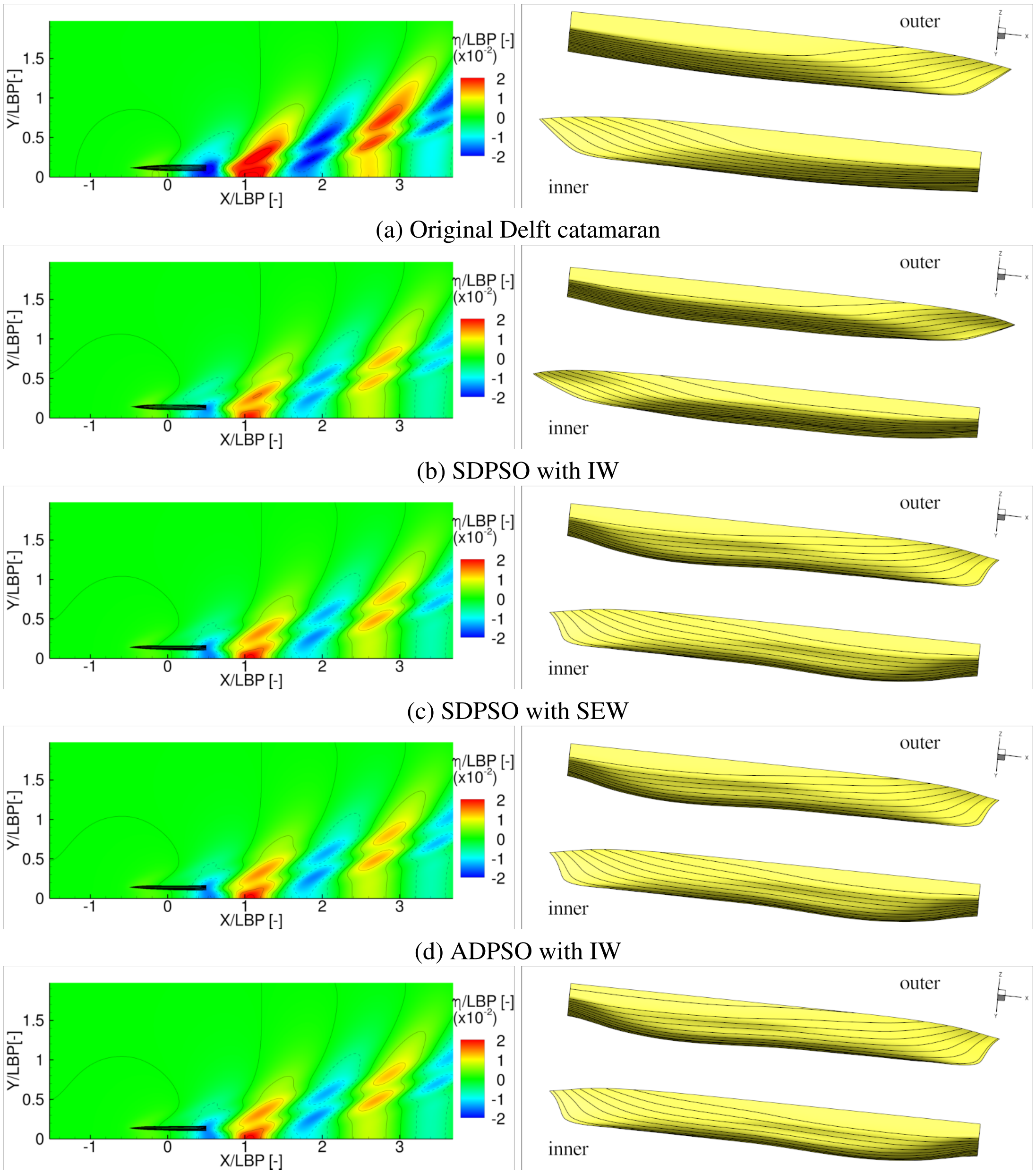
(b) SDPSO with IW

(c) SDPSO with SEW

(d) ADPSO with IW

(e) ADPSO with SEW

Figure 18: $4D$ SBD results.



(a) Original Delft catamaran

(b) SDPSO with IW

(c) SDPSO with SEW

(d) ADPSO with IW

(e) ADPSO with SEW

Figure 19: $6D$ SBD results.

6 Discussion

The analyses of SDPSO and ADPSO performances on the analytical test functions have shown the importance of the particle initialization especially for problems with a high number of design variables. In particular, the initialization on the only domain boundary is always the worst and should be avoided, whereas the use of a non-null initial velocity is always advisable (see Figs. 5b, 7b, 9b, and 11b).

The number of particles becomes moderately relevant when the budget of function evaluations increases, specially for problems with a smaller number of variables. More in detail, Figs. 5a, 7a, 9a, and 11a show that a higher number of particles performs better with a high budget of function evaluations. This can be explained recalling the relation between the number of particles and the number of iterations, for a fixed budget of function evaluations. On the one hand, a low number of particles requires a larger number of iterations, providing a fast convergence but possibly also a fast stagnation. On the other hand, a high number of particles requires a lower number of iterations, providing a slower convergence and possibly delaying stagnation. This is the reason why the number of particles should be determined considering the available budget, that is usually low in the SBD context, especially when CPU-time expensive solvers are used.

The coefficient set and the wall-type approach seem to be the less relevant parameters. Nevertheless, it can be said that Clercs coefficients ($\chi = 0.721$, $c_1 = c_2 = 1.655$ [31]) and the SEW-type approach are the most effective and efficient choices for both SDPSO and ADPSO (see Figs. 5c, 7c, 9c, 11c, 5d, 7d, 9d, and 11d). Moreover, the use of SEW is suggested in order to prevent an early stop of the swarm particle dynamics, as shown for the $6D$ design space of the naval engineering problem. The asynchronous mechanism of ADPSO shows equivalent or slightly better performance (in terms of number of objective function evaluations and objective reduction) compared to SDPSO, probably due to the diversity in the information update. This aspect of ADPSO provides a further interesting opportunity for the exploitation of parallel architectures in HPC systems.

7 Conclusions

A guideline for an effective and efficient use of SDPSO and ADPSO, in a computational framework characterized by limited resources, has been suggested. A parametric analysis 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 100 analytical test functions (with dimensionality from two to fifty) and three different absolute performance criteria. All possible combinations of DPSO parameters led to 420 optimizations for each function. The most promising DPSO setups have been discussed, identified, and successfully applied to a ship SBD optimization problem, namely the hull-form optimization of the high-speed Delft catamaran, advancing in calm water at fixed speed, using a potential-flow solver. The optimization aimed at the reduction of the ratio between the total resistance and the ship weight, using four- and six-dimensional design spaces. The particles initialization has been found the most significant parameter for the DPSO performance, especially for a number of design variables $N_{dv} \geq 10$ and low budgets of function evaluations. Conversely, the coefficients set and the wall-type approach have been found having a little influence on the DPSO performance, compared to the other parameters.

For problems with less then 10 variables, the suggested SDPSO and ADPSO setups follow: (a) number of particles N_p equal to 4 times the number of design variables; (b) particle initialization with HSS distribution on domain and bounds with non-null velocity; (c) set of coefficients proposed in [31], i.e., $\chi = 0.721$, $c_1 = c_2 = 1.655$; (d) inelastic and semi-elastic wall-type approach, for SDPSO and ADPSO respectively.

For problems with more then 10 design variables, the suggested setups for SDPSO and ADPSO follow: (a) number of particles N_p equal to 4 times the number of design variables; (b) particles initialization including HSS distribution on domain with non-null velocity; (c) set of coefficients proposed in [31]; (d) semi-elastic wall-type approach.

The performance of the guideline suggested for the test functions are very close or coincident to the best setup for each evaluation budget, among all 420 configurations available (see “Guide” and “Best” in Figs. 13 and 14). Finally, the guideline setups have proved to perform well on both the $4D$ and $6D$ ship SBD optimization problems.

Acknowledgments

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 Dr. Ki-Han Kim, and by the Italian Flagship Project RITMARE, coordinated by the Italian National Research Council and funded by the Italian Ministry of Education.

References

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

[2] Y. Zhang, S. Wang, G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, Mathematical Problems in Engineering 2015.

[3] 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.

[4] X. Chen, M. Diez, M. Kandasamy, E. F. 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, 2013.

[5] 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.

[6] 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.

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

[8] B. Rosenthal, D. Kring, Optimal design of a lifting body swath, in: Proceedings of 12th International Conference on Fast Sea Transportation (FAST2013), Amsterdam, The Netherlands, 2013.

[9] 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.

[10] 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.

[11] 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.

[12] A. Carlisle, G. Dozier, An off-the-shelf PSO, in: Proceding of the Workshop on Particle Swarm Optimization, 2001, pp. 1–6.

[13] 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.

[14] B. Koh, A. George, R. Haftka, B. Fregly, Parallel asynchronous particle swarm optimization, Int. J. Numer. Meth. Engng 67 (2006) 578–595.

[15] N. A. A. Aziz, Z. Ibrahim, Asynchronous particle swarm optimization for swarm robotics, Procedia Engineering: International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012) 41 (2012) 951 – 957.

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

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

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

[19] A. Engelbrecht, Particle swarm optimization: Velocity initialization, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.

[20] P. Cazzaniga, M. S. Nobile, D. Besozzi, The impact of particles initialization in PSO: Parameter estimation as a case in point, in: Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on, 2015, pp. 1–8.

[21] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, A. Abraham, Inertia weight strategies in particle swarm optimization, in: Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on, 2011, pp. 633–640.

[22] 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.

[23] A. Rezaee Jordehi, J. Jasni, Parameter selection in particle swarm optimisation: a survey, Journal of Experimental & Theoretical Artificial Intelligence 25 (4) (2013) 527–542.

[24] 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.

[25] E. F. Campana, G. Fasano, D. Peri, Penalty function approaches for Ship Multidisciplinary Design Optimization, European Journal of Industrial Engineering 6 (2012) 765–784.

[26] 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.

[27] 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.

[28] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimisation problems, International Journal of Mathematical Modelling and Numerical Optimisation 4.

[29] 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.

[30] T. T. Wong, W. S. Luk, P. A. Heng, Sampling with hammersley and halton points, Journal of Graphics Tools 2 (2) (1997) 9–24.

[31] 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`

[32] D. Peri, F. Tinti, A multistart gradient-based algorithm with surrogate model for global optimization, Communications in Applied and Industrial Mathematics 3 (1).

[33] 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.

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

[35] 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, 1999, pp. 1951–1957.

[36] 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.

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

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

[39] C. W. Cleghorn, A. P. Engelbrecht, A generalized theoretical deterministic particle swarm model, Swarm Intelligence 8 (1) (2014) 35–59.

[40] C. W. Cleghorn, A. P. Engelbrecht, Particle swarm variants: standardized convergence analysis, Swarm Intelligence 9 (2) (2015) 177–203.

[41] 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, Lecture Notes in Computer Science No. 8794, Springer International Publishing, 2014, pp. 126–133.

[42] K. Deb, An efficient constraint handling method for genetic algorithms, Computer Methods in Applied Mechanics and Engineering 186 (24) (2000) 311 – 338.

[43] M. Clerc, Confinements and biases in particle swarm optimization, Tech. rep. (2006). URL `http://clerc.maurice.free.fr/pso`

[44] D. Bratton, J. Kennedy, Defining a standard for particle swarm optimization, in: Proc. IEEE Swarm Intell, 2007, pp. 120–127.

[45] M. Corazza, G. Fasano, R. Gusso, Particle Swarm Optimization with non-smooth penalty reformulation, for a complex portfolio selection problem, Applied Mathematics and Computation 224 (2013) 611–624.

[46] A. Levy, A. Montalvo, S. Gomez, A. Galderon, Topics in Global Optimization, Lecture Notes in Mathematics No. 909, Springer-Verlag, 1981.

[47] H. Schlichting, K. Gersten, Boundary-Layer Theory, Springer-Verlag, Berlin, 2000.