

Dense Conjugate Initialization for Deterministic PSO in Applications: ORTHOinit+

Cecilia Leotardi · Andrea Serani · Matteo Diez ·
Emilio F. Campana · Giovanni Fasano · Riccardo
Gusso

the date of receipt and acceptance should be inserted later

Abstract This paper describes a class of novel initializations in Deterministic Particle Swarm Optimization (DPSO) for approximately solving costly unconstrained global optimization problems. The initializations are based on choosing specific dense initial positions and velocities for particles. These choices tend to induce in some sense orthogonality of particles' trajectories, in the early iterations, in order to better explore the search space. Our proposal is inspired by both a theoretical analysis on a reformulation of PSO iteration, and by possible limits of the proposals reported in [6,5]. We explicitly show that, in comparison with other initializations from the literature, our initializations tend to scatter PSO particles, at least in the first iterations. The latter goal is obtained by imposing that the initial choice of particles' position/velocity satisfies specific conjugacy conditions, with respect to a matrix depending on the parameters of PSO. In particular, by an appropriate condition on particles' velocities, our initializations also resemble and partially extend a general paradigm in the literature of exact methods for derivative-free optimization. Moreover, we propose dense initializations for DPSO, so that the final approximate global solution obtained is possibly not too sparse, which might cause troubles in some applications. Numerical results, on both Portfolio Selection and Computational Fluid Dynamics problems, validate our theory and prove the effectiveness of our proposal, which applies also in case different neighbourhood topologies are adopted in DPSO.

Keywords Global Optimization · Deterministic PSO (DPSO) · Particles initial position and velocity · Conjugate vectors

C. Leotardi, A. Serani, M. Diez
CNR-INM, National Research Council-Institute of Marine Engineering, Via di Vallerano 139, 00128, Rome, Italy
E-mail: (cecilia.leotardi, andrea.serani, matteo.diez)@cnr.it

E.F. Campana
CNR-DIITET, National Research Council-Engineering, ICT and Technology for Energy and Transportation Department, Piazzale Aldo Moro 7, 00185 Rome, Italy
E-mail: emiliofortunato.campana@cnr.it

G. Fasano
University Ca'Foscari of Venice, Department of Management, S.Giobbe, Cannaregio 873, 30121 Venice, Italy
E-mail: fasano@unive.it

R. Gusso
University Ca'Foscari of Venice, Department of Economics, S.Giobbe, Cannaregio 873, 30121 Venice, Italy
E-mail: rgusso@unive.it

1 Introduction

In this paper we first consider the solution of the unconstrained global optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is *continuous* and possibly *nondifferentiable*, so that derivatives are unavailable. In particular, we aim at detecting a global minimum \mathbf{x}^* of (1), satisfying $f(\mathbf{x}^*) \leq f(\mathbf{x})$, for any $\mathbf{x} \in \mathbb{R}^n$. As well known, the existence of a global minimum of (1) can be often guaranteed under mild assumptions on $f(\mathbf{x})$ (e.g., $f(\mathbf{x})$ is coercive, with $\lim_{\|\mathbf{x}\| \rightarrow \infty} f(\mathbf{x}) = +\infty$). Furthermore, in order to motivate the use of an heuristic procedure to solve (1), in place of possibly more expensive exact asymptotic convergent methods, we also assume that the function $f(\mathbf{x})$ is computationally *expensive* and a fast approximate solution is sought. Observe that in the last decade we have seen a fast increasing complexity in applications, where optimization techniques are sought [18, 22, 44]. In particular, the demand of sophisticated and efficient methods which do not rely on the smoothness of the functions has considerably speeded up, because of the abundance of problems where derivatives are unavailable and the functions are only accessible through black-boxes (see [15]). In this regard, Automatic Differentiation [23] helps to exactly retrieve derivatives using only function evaluations, but it needs the analytical expression of the functions in hand. Conversely, two main derivative-free (DFO) approaches have gained the attention in the last decades, both with *pros* and *cons*. Direct Search methods, which exploit the objective function information along a precise pattern of search directions (see [27]), and Model Based DFO methods, which use information of the function based on a local model (see [15]). In particular, among recent advances we can find the class of Mesh Adaptive Direct Search (MADS) algorithms [1, 2], along with their variants. They represent a natural evolution of Direct Search methods for nonsmooth constrained problems, and are based on considering a dense set of search directions.

To our purposes in this paper we focus on the heuristic procedure PSO, which is neither able to guarantee the convergence to global minima nor it ensures global convergence (i.e. convergence from any starting iterate) to stationary points. PSO represents an iterative method for the approximate solution of global optimization problems, based on updating a swarm of P *particles* (see [26, 4, 11, 12] and therein references). The use of heuristic methods in place of DFO globally convergent techniques is advisable in those cases where the function evaluation is rather expensive, or the number of function evaluations, necessary for the solution of the problem in hand, tends to increase. In this regard, heuristics can offer a reasonable compromise between the accuracy of the final solution and the computational budget required to obtain it. Each particle represents a vector (position) in \mathbb{R}^n and iteratively describes a trajectory in \mathbb{R}^n . At the k -th iteration the j -th particle updates its position as

$$\mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \mathbf{v}_j^{k+1}, \quad j = 1, \dots, P, \quad (2)$$

where $\mathbf{x}_j^k \in \mathbb{R}^n$ indicates its current *position*, $\mathbf{v}_j^{k+1} \in \mathbb{R}^n$ is its *velocity* and represents a search direction. Borrowing a standard terminology from optimization, we have in general that \mathbf{v}_j^{k+1} is not *gradient related*. In other words, \mathbf{v}_j^{k+1} might possibly be not a descent direction for $f(\mathbf{x})$ at \mathbf{x}_j^k , so that any small movement along \mathbf{v}_j^{k+1} might not yield a function decrease. As a natural consequence, we have that as already remarked, the *global convergence properties* for PSO iteration (2) can be hardly proved without further additional assumptions. Observe that some papers have been proposed in the literature, where PSO was paired with derivative-free globally convergent methods, based on both linesearch methods (see [41, 39]) and pattern search methods (see [49, 48]). In the current paper our aim is limited to the study of favorable variants of PSO initializations, without issuing problems related to global convergence.

In PSO, the original expression of vector \mathbf{v}_j^{k+1} (see also [26]), for a particle j , was given by

$$\mathbf{v}_j^{k+1} = \mathbf{v}_j^k + c_j \mathbf{r}_j \otimes (\mathbf{p}_j^k - \mathbf{x}_j^k) + c_g \mathbf{r}_g \otimes (\mathbf{p}_g^k - \mathbf{x}_j^k), \quad (3)$$

where c_j and c_g are constant parameters, while \mathbf{r}_j and \mathbf{r}_g are suitable random vectors. Finally, the symbol ‘ \otimes ’ represents a component-wise product, and

$$\begin{aligned} \mathbf{p}_j^k &\in \operatorname{argmin}_{0 \leq h \leq k} \{f(\mathbf{x}_j^h)\}, & j = 1, \dots, P, \\ \mathbf{p}_g^k &\in \operatorname{argmin}_{0 \leq h \leq k, j=1, \dots, P} \{f(\mathbf{x}_j^h)\}, \end{aligned} \quad (4)$$

being, at step k , \mathbf{p}_j^k the *personal best* position so far of particle j , and \mathbf{p}_g^k the *global best* position so far of the entire swarm, respectively.

Relations (2)-(3) reveal that in PSO iteration, the current position of a particle depends on its lattermost position, but is also independent of the other previous positions. Equivalently, PSO can be viewed as a Markov chain. As usually reported in the literature, c_j is the so called *cognitive parameter*, and multiplies the contribution from the history of the j -th particle. On the other hand, c_g is the *social parameter*, and the vector $c_g \mathbf{r}_g \otimes (\mathbf{p}_g^k - \mathbf{x}_j^k)$ in (3) attempts to balance the information from each particle’s history through a social contribution, summarized by the current global best position in the swarm.

This paper is mainly focused on addressing the initialization in PSO (for a partial analysis and a numerical experience on the impact of PSO initialization, the reader can refer to the recent paper by [42] and therein references). Specifically, we first consider a DPSO iteration as in [6]. From the latter paper we also borrow the idea of routing the particles’ trajectories so that they possibly follow nearly orthogonal patterns, in a sense which is specified in Section 4. However, in this paper we also attempt to overcome some limits in [6] and [20], as indicated in the following items.

- (i) In [6] the authors tend to impose (at least in the early DPSO iterations) the orthogonality of the entire *trajectories* of the particles, in an extended search space. On the contrary, here we provide analytical conditions to impose only orthogonal *velocities* of particles, in the early iterations. The resulting approach proves to be more efficient than the one in [6].
- (ii) Similarly to [20] we first tend to impose the orthogonality of particles’ velocities, in the early iterations. Then, unlike in [20] we suitably recombine vectors obtained by solving an intermediate symmetric eigenvalue problem, so that linear independency among the velocities is finally pursued, at least in the early iterations.
- (iii) We can guarantee that, unlike [6] and [20], our proposal here introduces ‘dense’ approximations for the initial positions and velocities of particles. I.e., we guarantee that particles initialization is not sparse, which may cause on some applications an undesired bias of the final results (see also Section 5).

On the other hand, our approach takes into account the results obtained in the landmark papers [26, 34, 14, 32, 36], along with [46, 45, 47], where several relevant issues related to PSO initialization are investigated. Note that in our numerical experience here, we are not concerned with comparing our proposal with other efficient initializations from the literature. This choice is motivated by one fact, which definitely makes the latter comparison unfair. On one hand, more standard initializations typically handle random positions and velocities, so that they require a statistical analysis in order to validate the results they provide. This turns to increase the overall number of function evaluations, which might compromise efficiency. Since our proposal does not use any random parameter, to some extent we might be possibly less flexible, however no statistics on the numerical results is needed (see also [42, 38]). The latter fact turns to be an essential requisite on our ship design problems, where each objective function evaluation is often the result of a time consuming simulation. Hence, we preferred to report our numerical experience in Section 7 including only *deterministic* initializations paired with DPSO.

For the sake of completeness, we also highlight that our deterministic initialization can be hardly paired with *non-deterministic* versions of PSO. Indeed, the presence of random parameters in PSO might cause the matrix $\mathbf{Q}(k)$ in (9) to yield the inequality $\mathbf{Q}(k) \neq [\mathbf{Q}(1)]^k$, which destroys the achievements in Sections 3 and 4.

Sections 2 and 3 briefly report a revised version of the proposal by [6], while Section 4 contains both the first contribution of this paper, along with some theoretical results. Sections 5 and 6 motivate a more specific initialization for DPSO, which is then tested on a ship design problem in Section 7. Finally, Section 8 and an Appendix will complete the paper, including some extensions and technical proofs.

We recall that with ‘ $\det(\mathbf{A})$ ’ we indicate the determinant of matrix \mathbf{A} . ‘ \mathbf{I} ’ indicates the identity matrix of suitable dimension and ‘ \mathbf{e}_i ’ is the i -th unit vector. To avoid a burdensome notation the Euclidean norm is simply indicated by $\|\cdot\|$, while $\|\cdot\|_p$ represents the standard p -norm.

2 PSO iteration as a linear dynamic system

In order to describe our novel initialization for a modified PSO scheme, let us consider the next Assumption 1 (which characterizes the so called Deterministic Particle Swarm Optimization - DPSO, see also [35]).

Assumption 1 [DPSO] *We assume in (3) that $c_j = c$ and $\mathbf{r}_j = \mathbf{r}$, for any $j = 1, \dots, P$. Moreover, we set $c_g = \bar{c}$, $\mathbf{r}_g = \bar{\mathbf{r}}$, being $c, \mathbf{r}, \bar{c}, \bar{\mathbf{r}}$ suitable positive constant coefficients.*

Now, following the guidelines by [42], we also introduce in (3) the real parameter χ (*constriction factor*) and choose $\mathbf{r} = \bar{\mathbf{r}} = \mathbf{1}$ in Assumption 1. Then, we may consider for the j -th particle the overall iteration of DPSO:

$$\begin{cases} \mathbf{v}_j^{k+1} = \chi [\mathbf{v}_j^k + c_j(\mathbf{p}_j^k - \mathbf{x}_j^k) + c_g(\mathbf{p}_g^k - \mathbf{x}_j^k)], & k \geq 0, \\ \mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \mathbf{v}_j^{k+1}, & k \geq 0. \end{cases} \quad (5)$$

Without loss of generality at present we focus on the j -th particle and omit the subscript in the recurrence (5), so that for the sake of simplicity we can set $\mathbf{p}_j^k = \mathbf{p}^k$, $\mathbf{x}_j^k = \mathbf{x}^k$ and $\mathbf{v}_j^k = \mathbf{v}^k$.

With the latter position the iteration (5) is immediately equivalent to the *dynamic, linear and stationary system*¹

$$\mathbf{X}(k+1) = \begin{bmatrix} \chi \mathbf{I} & -\chi(c + \bar{c}) \mathbf{I} \\ \chi \mathbf{I} & [1 - \chi(c + \bar{c})] \mathbf{I} \end{bmatrix} \mathbf{X}(k) + \begin{bmatrix} \chi(c\mathbf{p}^k + \bar{c}\mathbf{p}_g^k) \\ \chi(c\mathbf{p}^k + \bar{c}\mathbf{p}_g^k) \end{bmatrix}, \quad (6)$$

where c and \bar{c} are defined in Assumption 1 and

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{v}^k \\ \mathbf{x}^k \end{bmatrix} \in \mathbb{R}^{2n}, \quad k \geq 0.$$

The sequence $\{\mathbf{X}(k)\}$ identifies a trajectory in the state space \mathbb{R}^{2n} , and since (6) is a linear and stationary system, we may consider the *free response* $\mathbf{X}_L(k)$ and the *forced response* $\mathbf{X}_F(k)$ of the trajectory $\{\mathbf{X}(k)\}$. Then, considering (6) we explicitly obtain at step $k \geq 0$ (see also [6,5])

$$\mathbf{X}(k) = \mathbf{X}_L(k) + \mathbf{X}_F(k), \quad (7)$$

where

$$\mathbf{X}_L(k) = \mathbf{Q}(k)\mathbf{X}(0), \quad \mathbf{X}_F(k) = \sum_{\tau=0}^{k-1} \mathbf{H}(k-\tau)\mathbf{U}(\tau), \quad (8)$$

¹ The latter terminology is borrowed from [6,5], whose symbols are used in this brief section, too.

and after some calculations, we obtain for the quantities in (8) the expressions

$$\mathbf{Q}(k) = \begin{bmatrix} \chi \mathbf{I} & -\chi(c + \bar{c}) \mathbf{I} \\ \chi \mathbf{I} & [1 - \chi(c + \bar{c})] \mathbf{I} \end{bmatrix}^k \in \mathbb{R}^{2n \times 2n}, \quad (9)$$

$$\mathbf{H}(k - \tau) = \begin{bmatrix} \chi \mathbf{I} & -\chi(c + \bar{c}) \mathbf{I} \\ \chi \mathbf{I} & [1 - \chi(c + \bar{c})] \mathbf{I} \end{bmatrix}^{k - \tau - 1} \in \mathbb{R}^{2n \times 2n}, \quad (10)$$

$$\mathbf{U}(\tau) = \begin{bmatrix} \chi(c \mathbf{p}^k + \bar{c} \mathbf{p}_g^k) \\ \chi(c \mathbf{p}^k + \bar{c} \mathbf{p}_g^k) \end{bmatrix} \in \mathbb{R}^{2n}, \quad (11)$$

being the matrices $\mathbf{Q}(k)$, $\mathbf{H}(k - \tau)$ and $\mathbf{U}(\tau)$ in general dependent on the iteration index k . As detailed hereafter, for our purposes this last dependency does not represent a drawback, inasmuch as Assumption 1 holds and $\mathbf{Q}(k)$ in (9) is the power of a constant matrix. Also observe that in (9) the matrix $\mathbf{Q}(k)$ explicitly accounts for the dependency of $\mathbf{X}(k)$ on $\mathbf{X}(0)$ (and not on the vectors \mathbf{p}^k , \mathbf{p}_g^k). On the other hand, the matrices $\mathbf{H}(k - \tau)$ and $\mathbf{U}(\tau)$ uniquely depend on \mathbf{p}^k , \mathbf{p}_g^k , i.e. they depend on the knowledge collected during the progress of the algorithm, and do not depend on the initial condition $\mathbf{X}(0)$. Thus, $\mathbf{H}(k - \tau)$ and $\mathbf{U}(\tau)$ uniquely account for the dependency of $\mathbf{X}(k)$ on \mathbf{p}^k , \mathbf{p}_g^k .

We want to specifically focus now on the free response $\mathbf{X}_L(k)$, so that we need to preliminarily study the main features of $\mathbf{Q}(k)$. A remarkable observation from the latter formulae is that $\mathbf{X}_L(k)$ in (8) *uniquely depends on the initial point* $\mathbf{X}(0)$, being independent of the vector \mathbf{p}_g^k . On the contrary, $\mathbf{X}_F(k)$ in (8) is independent of $\mathbf{X}(0)$, while it is strongly dependent on \mathbf{p}_g^k . This implies that the quantities $\mathbf{X}_L(k)$ and $\mathbf{X}_F(k)$ can be separately computed. For our analysis we need now to explicitly calculate the eigenpairs of $\mathbf{Q}(k)$, in order to easily compute the free response $\mathbf{X}_L(k)$.

3 Computation of the free response $\mathbf{X}_L(k)$ and choice of DPSO coefficients

Suppose that Assumption 1 holds, hereafter in order to simplify the notation we introduce the following position in (9)

$$\omega = \chi(c + \bar{c}). \quad (12)$$

Then, before proceeding with our analysis, we recall (see also [46, 14, 45, 47, 6]) that in order to provide necessary conditions which avoid divergence of the trajectories of particles, the relations

$$\begin{aligned} 0 < \chi < 1 \\ 0 < \omega < 2(\chi + 1) \end{aligned} \quad (13)$$

must hold. In this regard other settings for DPSO parameters can be chosen, as specified by [47]. In any case, the relations (13) guarantee that all the eigenvalues of $\mathbf{Q}(k)$ have a modulus smaller than one; moreover, relations (13) also ensure that

$$\lim_{k \rightarrow \infty} \mathbf{X}_L(k) = 0$$

by the definition of free response. A keynote fact we use in this paper is that (see [6]) the matrix $\mathbf{Q}(1)$ in (8) has just the two real eigenvalues λ_1 and λ_2 if

$$\omega \leq 1 + \chi - 2\sqrt{\chi} \quad \text{or} \quad \omega \geq 1 + \chi + 2\sqrt{\chi},$$

being λ_1 and λ_2 coincident if and only if $\omega = (1 \pm \sqrt{\chi})^2$. In case $\lambda_1 = \lambda_2$ then some care has to be considered in our analysis, which is beyond the scope of the present paper. Thus, we assume that the next condition holds, too.

Assumption 2 *Let Assumption 1 hold. Given the quantities χ and ω in (12), we also assume that*

$$\omega \neq (1 \pm \sqrt{\chi})^2.$$

If Assumption 1 holds, the results by [6] can be applied, so that the free response at step k , i.e. $\mathbf{X}_L(k) = \mathbf{Q}(k)\mathbf{X}(0) = [\mathbf{Q}(1)]^k \mathbf{X}(0)$, can be computed by simply using the distinct eigenvalues λ_1 and λ_2 of $\mathbf{Q}(1)$, being explicitly

$$\mathbf{X}_L(k) = [\mathbf{Q}(1)]^k \mathbf{X}(0) = \begin{bmatrix} \gamma_1(k)\mathbf{v}^0 - \gamma_2(k)\mathbf{x}^0 \\ \gamma_3(k)\mathbf{v}^0 - \gamma_4(k)\mathbf{x}^0 \end{bmatrix}, \quad (14)$$

where, after some computations

$$\begin{aligned} \gamma_1(k) &= \frac{\lambda_1^k(\chi - \lambda_2) - \lambda_2^k(\chi - \lambda_1)}{\lambda_1 - \lambda_2}; & \gamma_2(k) &= \frac{\omega(\lambda_1^k - \lambda_2^k)}{\lambda_1 - \lambda_2}; \\ \gamma_3(k) &= \frac{(\lambda_1^k - \lambda_2^k)(\chi - \lambda_1)(\chi - \lambda_2)}{\omega(\lambda_1 - \lambda_2)}; & \gamma_4(k) &= \frac{\lambda_1^k(\chi - \lambda_1) - \lambda_2^k(\chi - \lambda_2)}{\lambda_1 - \lambda_2}. \end{aligned}$$

4 An attempt to improve the effectiveness for our DPSO initialization

This section is mainly devoted to study a possible improvement for DPSO initialization, provided that Assumptions 1 and 2 hold. Our proposal allows DPSO to

1. widely exploring the search space in the early iterations,
2. maintaining iteration (5),
3. possibly reducing the overall computational effort, when measured throughout the number of function evaluations.

Note that as regards item 2., we are interested about *improving* DPSO basic iteration, rather than modifying it. That is why our proposal is intended as an enhancement of an existing method, rather than a proposal for a novel one. As regards item 3., it may have a dramatic impact on some applications, where each function evaluation requires a large CPU-time (e.g. design problems, as better detailed in Section 7.2). On this purpose, we strongly remark that exploiting our initializations both

- a few iterations of DPSO might potentially suffice to provide a reasonable and satisfactory approximate solution of (1);
- the final solution provided is possibly dense, i.e. a large number of its entries are nonzero.

We start our analysis using the reformulation in Section 2, in order to impose a novel condition (with respect to [6]) for the choice of initial particles' position/velocity (namely the next relation (18)). Then, in Section 6 we will suitably update (18) in order to avoid sparsity on the final approximate solution.

To sketch our proposal we consider any two particles j and h , such that $1 \leq j \neq h \leq P$. Using the theory in Section 2, along with the Assumptions 1 and 2, we consider their trajectories in the space \mathbb{R}^{2n} , so that their initial position ($\mathbf{X}(0)^{(j)}$ and similarly $\mathbf{X}(0)^{(h)}$) and free response ($\mathbf{X}_L(k)^{(j)}$ and similarly $\mathbf{X}_L(k)^{(h)}$) are given by (14) (we only report results for particle j , but similar expressions immediately hold also for particle h)

$$\mathbf{X}(0)^{(j)} = \begin{bmatrix} \mathbf{v}_j^0 \\ \mathbf{x}_j^0 \end{bmatrix} \implies \mathbf{X}_L(k)^{(j)} = [\mathbf{Q}(1)]^k \mathbf{X}(0)^{(j)} = \begin{bmatrix} \gamma_1(k)\mathbf{v}_j^0 - \gamma_2(k)\mathbf{x}_j^0 \\ \gamma_3(k)\mathbf{v}_j^0 - \gamma_4(k)\mathbf{x}_j^0 \end{bmatrix}. \quad (15)$$

Borrowing the idea in Section 6 of [6], for a given value of the index k we might consider to impose the orthogonality of the free responses $\{\mathbf{X}_L(k)^{(j)}\}$ (and not necessarily the orthogonality of

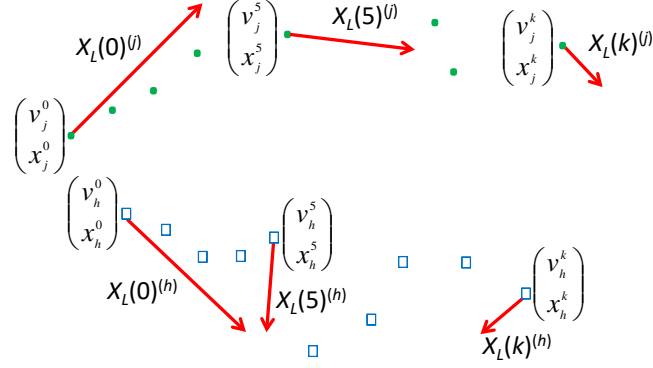


Fig. 1 Green points and blue squares represent trajectories of the particles j and h , respectively, in the extended space \mathbb{R}^{2n} . [6] impose the orthogonality between the (entire) free responses $\mathbf{X}_L(k)^{(j)}$ and $\mathbf{X}_L(k)^{(h)}$, in order to improve the exploration of the search space.

the vectors $\{\mathbf{X}(k)^{(j)}\}$ - see Figure 1. This is not equivalent, of course, to impose the orthogonality of particles' trajectories. Nevertheless, as the numerical experience in [6] confirms, we observed an appreciable efficiency by setting the position and velocity of $P \equiv n$ particles, in such a way that the free responses $\mathbf{X}_L(k)^{(j_1)}, \dots, \mathbf{X}_L(k)^{(j_n)}$ possibly satisfy at step k the $P \cdot (P - 1)$ orthogonality conditions

$$\left[\mathbf{X}_L(k)^{(j_i)}\right]^T \left[\mathbf{X}_L(k)^{(j_h)}\right] = 0, \quad \forall j_i, j_h \in \{j_1, \dots, j_n\}, \quad i \neq h. \quad (16)$$

Since the free responses of particles also satisfy, for any particle j , the condition

$$\lim_{k \rightarrow \infty} \mathbf{X}_L(k)^{(j)} = 0,$$

i.e. they tend to fade with the iterations, then it was reasonable in [6] to impose the orthogonality conditions (16) at step $k = 0$, i.e. at the outset of the iterative process.

Here, we want to generalize the idea in [6], observing that in order to scatter the particles in the search space what really matters is the orthogonality of the search directions of the particles, and possibly not the orthogonality of the entire free responses as in (16). On this guideline, here we want to study the initial position and velocity of $2n$ particles, so that at step $k = 0$ (i.e. at the very early iteration) the corresponding free responses $\mathbf{X}_L(0)^{(j_1)}, \dots, \mathbf{X}_L(0)^{(j_{2n})}$ satisfy for any $1 \leq j \neq h \leq 2n$ (see (15))

$$[\gamma_1(1)\mathbf{v}_j^0 - \gamma_2(1)\mathbf{x}_j^0]^T [\gamma_1(1)\mathbf{v}_h^0 - \gamma_2(1)\mathbf{x}_h^0] = 0. \quad (17)$$

I.e., only the components corresponding to the velocity, in the free responses of particles j and h , are orthogonal. After some computation, relation (17) is equivalent to impose the initial condition

$$\left[\mathbf{X}(0)^{(j)}\right]^T \begin{bmatrix} \sigma_1 \mathbf{I} & \sigma_2 \mathbf{I} \\ \sigma_2 \mathbf{I} & \sigma_3 \mathbf{I} \end{bmatrix} \mathbf{X}(0)^{(h)} = 0, \quad (18)$$

where

$$\sigma_1 = [\gamma_1(1)]^2; \quad \sigma_2 = -\gamma_1(1)\gamma_2(1); \quad \sigma_3 = [\gamma_2(1)]^2.$$

Observe that the matrix

$$\mathbf{A} = \begin{bmatrix} \sigma_1 \mathbf{I} & \sigma_2 \mathbf{I} \\ \sigma_2 \mathbf{I} & \sigma_3 \mathbf{I} \end{bmatrix} \quad (19)$$

is symmetric and by (15) the condition (18) indicates that the vectors

$$\mathbf{X}(0)^{(j)}, \quad \mathbf{X}(0)^{(h)} \quad (20)$$

must be *mutually \mathbf{A} -conjugate* (see also [25,21] for a reference on conjugacy and its applications in optimization).

4.1 Some theoretical results

The first relevant property induced by the introduction of the \mathbf{A} -conjugacy in relation (18) is the linear independence, as stated in the next result.

Proposition 1 *Let $m \leq 2n$ and consider relations (15). Suppose the vectors*

$$\mathbf{X}(0)^{(j)} \in \mathbb{R}^{2n}, \quad j = 1, \dots, m, \quad (21)$$

are mutually conjugate with respect to the symmetric non singular matrix $\mathbf{A} \in \mathbb{R}^{2n \times 2n}$ in (19), i.e.

$$\left[\mathbf{X}(0)^{(i)} \right]^T \mathbf{A} \mathbf{X}(0)^{(j)} = 0, \quad 1 \leq i \neq j \leq m,$$

with

$$\left[\mathbf{X}(0)^{(j)} \right]^T \mathbf{A} \mathbf{X}(0)^{(j)} \neq 0, \quad 1 \leq j \leq m. \quad (22)$$

Then, the vectors (21) are linearly independent in \mathbb{R}^{2n} .

Proof

We directly have by the \mathbf{A} -conjugacy and conditions (22) that

$$\begin{aligned} \det \left[(\mathbf{X}(0)^{(1)} \dots \mathbf{X}(0)^{(m)})^T \mathbf{A} (\mathbf{X}(0)^{(1)} \dots \mathbf{X}(0)^{(m)}) \right] = \\ \det \left[\text{diag}_{1 \leq j \leq m} \left\{ \left[\mathbf{X}(0)^{(j)} \right]^T \mathbf{A} \mathbf{X}(0)^{(j)} \right\} \right] \neq 0, \end{aligned}$$

which follows from (22). Thus, the vectors in (21) are linearly independent. \diamond

Now, the property in Proposition 1 suggests that in case the vectors (21) are mutually conjugate, then not only the velocities of the free responses of the particles, at step $k = 1$, are orthogonal (as stated in relation (17)), but the vectors (21) will be also *sufficiently well* scattered in \mathbb{R}^{2n} (since they are linearly independent). Thus, the choice (21) also underlies a distribution of points in the extended space \mathbb{R}^{2n} .

To complete our analysis we need now to provide an automatic and reliable procedure for generating the mutually \mathbf{A} -conjugate vectors (21), possibly having $m = 2n$, being $2n$ the largest possible number of \mathbf{A} -conjugate (and linearly independent) vectors in \mathbb{R}^{2n} satisfying Proposition 1. To this purpose we have the following result.

Proposition 2 *Let $m \leq 2n$ and suppose the matrix $\mathbf{A} \in \mathbb{R}^{2n \times 2n}$ in (19) is symmetric positive definite. Let*

$$\mathbf{X}(0)^{(1)} \in \mathbb{R}^{2n}$$

be any vector such that

$$\left[\mathbf{X}(0)^{(1)} \right]^T \mathbf{A} \mathbf{X}(0)^{(1)} \neq 0, \quad (23)$$

and let us set the parameter $\varepsilon_1 \in \mathbb{R}$ such that

$$\varepsilon_1 = \frac{\|\mathbf{v}_1^0\|^2 + \|\mathbf{x}_1^0\|^2}{\sigma_1 \|\mathbf{v}_1^0\|^2 + \sigma_3 \|\mathbf{x}_1^0\|^2 + 2\sigma_2 (\mathbf{v}_1^0)^T (\mathbf{x}_1^0)}, \quad (24)$$

and the vectors $\mathbf{s}_1^0 \in \mathbb{R}^n$, $\mathbf{t}_1^0 \in \mathbb{R}^n$ such that

$$\begin{bmatrix} \mathbf{s}_1^0 \\ \mathbf{t}_1^0 \end{bmatrix} = \mathbf{X}(0)^{(1)}. \quad (25)$$

Suppose $\sigma_1 \|\mathbf{v}_j^0\|^2 + \sigma_3 \|\mathbf{x}_j^0\|^2 + 2\sigma_2 (\mathbf{v}_j^0)^T (\mathbf{x}_j^0) \neq 0$, for $j = 1, \dots, m$, and consider the recurrences

$$\begin{bmatrix} \mathbf{s}_j^0 \\ \mathbf{t}_j^0 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{j-1}^0 - \varepsilon_{j-1} (\sigma_1 \mathbf{v}_{j-1}^0 + \sigma_2 \mathbf{x}_{j-1}^0) \\ \mathbf{t}_{j-1}^0 - \varepsilon_{j-1} (\sigma_3 \mathbf{x}_{j-1}^0 + \sigma_2 \mathbf{v}_{j-1}^0) \end{bmatrix}, \quad j = 2, \dots, m, \quad (26)$$

$$\eta_j = \frac{\sigma_1 (\mathbf{s}_j^0)^T (\mathbf{v}_{j-1}^0) + \sigma_2 [(\mathbf{t}_j^0)^T (\mathbf{v}_{j-1}^0) + (\mathbf{s}_j^0)^T (\mathbf{x}_{j-1}^0)] + \sigma_3 (\mathbf{t}_j^0)^T (\mathbf{x}_{j-1}^0)}{\sigma_1 \|\mathbf{v}_{j-1}^0\|^2 + \sigma_3 \|\mathbf{x}_{j-1}^0\|^2 + 2\sigma_2 (\mathbf{v}_{j-1}^0)^T (\mathbf{x}_{j-1}^0)}, \quad j = 2, \dots, m, \quad (27)$$

$$\mathbf{X}(0)^{(j)} = \begin{bmatrix} \mathbf{s}_j^0 - \eta_j \mathbf{v}_{j-1}^0 \\ \mathbf{t}_j^0 - \eta_j \mathbf{x}_{j-1}^0 \end{bmatrix}, \quad j = 2, \dots, m, \quad (28)$$

$$\varepsilon_j = \frac{\|\mathbf{s}_j^0\|^2 + \|\mathbf{t}_j^0\|^2}{\sigma_1 \|\mathbf{v}_j^0\|^2 + \sigma_3 \|\mathbf{x}_j^0\|^2 + 2\sigma_2 (\mathbf{v}_j^0)^T (\mathbf{x}_j^0)}, \quad j = 2, \dots, m, \quad (29)$$

in order to generate the sequence of m vectors

$$\left\{ \mathbf{X}(0)^{(j)} \right\}_{k=1, \dots, m}. \quad (30)$$

Then, the m vectors in (30) are mutually \mathbf{A} -conjugate.

Proof

The proof follows after some computations, applying the Conjugate Gradient method in [25]. For the sake of brevity the proof is omitted. \diamond

Observe that Proposition 2 provides an efficient general iterative tool in order to generate the mutually \mathbf{A} -conjugate vectors

$$\mathbf{X}(0)^{(j)}, \quad j = 1, \dots, m,$$

also considering that the linear algebra involved is reasonably cheap, even for large values of m .

Remark 1 It is not difficult (though tedious) to verify that the recurrence (23)-(29) is to some extent equivalent to apply the so called method of Conjugate Gradient (CG, [25]), for the solution of the linear system

$$\mathbf{B}\mathbf{w} = \mathbf{b}, \quad \text{where} \quad \mathbf{B} = \mathbf{A}, \quad \mathbf{b} = - \begin{bmatrix} \mathbf{v}_1^0 \\ \mathbf{x}_1^0 \end{bmatrix}.$$

The latter method is designed for *symmetric positive definite* linear systems and generates the sequence $\{\mathbf{w}_k\}$ of approximations to the solution \mathbf{w}^* . The CG is widely used to solve linear systems within continuous optimization problems, since it is very stable and computationally cheap, provided that the condition number of matrix \mathbf{A} is reasonably bounded. However, note that in our case what really matters is the generation of the sequence (30), and not the pure investigation of possible solutions for the linear system whose matrix is \mathbf{A} . Moreover, the main assumption in Proposition 2 (i.e. the positive definiteness of matrix \mathbf{A}) unfortunately cannot hold, as detailed in the next section.

4.2 A further improvement

After a careful reading of Proposition 2, we can observe the following couple of weaknesses which possibly discourage the application of therein procedure, in order to compute the sequence (30):

- (i) Proposition 2 assumes that m mutually conjugate vectors are generated, but it does not guarantee that $m = 2n$. Thus, the procedure (23)-(29) generates a number of conjugate vectors which is possibly inferior to $2n$ (by Proposition 1 it cannot exceed $2n$), so that the advantages provided by imposing condition (17) might be not fully exploited;
- (ii) the denominator in the expression of the parameters $\{\varepsilon_j\}$ and $\{\eta_j\}$ in (24), (27) and (29) might be possibly zero, since in general the matrix \mathbf{A} has not all positive eigenvalues. Indeed, first observe that since \mathbf{A} is symmetric, its eigenvalues are all real. Moreover, we have for any $\lambda \neq \sigma_1$

$$[\mathbf{A} - \lambda \mathbf{I}] = \begin{bmatrix} \sigma_1 \mathbf{I} - \lambda \mathbf{I} & \sigma_2 \mathbf{I} \\ \sigma_2 \mathbf{I} & \sigma_3 \mathbf{I} - \lambda \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \frac{\sigma_2}{\sigma_1 - \lambda} \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} (\sigma_1 - \lambda) \mathbf{I} & \sigma_2 \mathbf{I} \\ \mathbf{0} & \left(\sigma_3 - \lambda - \frac{\sigma_2^2}{\sigma_1 - \lambda} \right) \mathbf{I} \end{bmatrix},$$

so that σ_1 cannot be an eigenvalue of \mathbf{A} and we have for the secular equation

$$0 = \det \begin{bmatrix} \sigma_1 \mathbf{I} - \lambda \mathbf{I} & \sigma_2 \mathbf{I} \\ \sigma_2 \mathbf{I} & \sigma_3 \mathbf{I} - \lambda \mathbf{I} \end{bmatrix} = (\sigma_1 - \lambda)^n \left[\sigma_3 - \lambda - \frac{\sigma_2^2}{\sigma_1 - \lambda} \right]^n,$$

which is satisfied if and only if

$$\lambda^2 - (\sigma_1 + \sigma_3)\lambda + (\sigma_1\sigma_3 - \sigma_2^2) = 0.$$

Thus, we obtain for the $2n$ eigenvalues of the (symmetric) matrix in (19) only the two (real) distinct values

$$\lambda_{1,2} = \frac{1}{2} \left[(\sigma_1 + \sigma_3) \pm \sqrt{(\sigma_1 + \sigma_3)^2 - 4(\sigma_1\sigma_3 - \sigma_2^2)} \right].$$

Now, recalling that

$$\sigma_1 = [\gamma_1(1)]^2; \quad \sigma_2 = -\gamma_1(1)\gamma_2(1); \quad \sigma_3 = [\gamma_2(1)]^2$$

we finally obtain the two eigenvalues

$$\begin{cases} \lambda_1 = 0 \\ \lambda_2 = [\gamma_1(1)]^2 + [\gamma_2(1)]^2 > 0, \end{cases}$$

each with algebraic multiplicity equal to n . Hence, the matrix \mathbf{A} is *singular* and only *positive semidefinite*, implying that the procedure in Proposition 2 can stop prematurely, with m possibly much smaller than $2n$.

Nevertheless, to overcome the disadvantages in (i)-(ii), note that if \mathbf{z}_i and \mathbf{z}_j are distinct eigenvectors of matrix \mathbf{A} , respectively associated to the eigenvalues λ_i and λ_j , then we simply have

$$\mathbf{z}_i^T \mathbf{A} \mathbf{z}_j = \mathbf{z}_i^T (\lambda_j \mathbf{z}_j) = \lambda_j \mathbf{z}_i^T \mathbf{z}_j = 0. \quad (31)$$

Thus, the eigenvectors of a symmetric matrix are also mutually conjugate directions with respect to that matrix. As a consequence, in order to satisfy condition (18) it suffices to compute the $2n$ eigenvectors $\mathbf{z}_1, \dots, \mathbf{z}_{2n}$ of (19) and set the vectors

$$\mathbf{X}(0)^{(1)}, \dots, \mathbf{X}(0)^{(2n)}$$

as proportional to the latter eigenvectors.

After some computation we have for the corresponding $2n$ eigenvectors $\mathbf{z}_i, \mathbf{z}_{n+i}, i = 1, \dots, n$, of the matrix \mathbf{A} in (19) the simple expressions

$$\lambda_1 \implies \mathbf{z}_i = \begin{bmatrix} -\frac{\sigma_3 - \lambda_1}{\sigma_2} \mathbf{e}_i \\ \mathbf{e}_i \end{bmatrix} = \begin{bmatrix} \frac{\gamma_2(0)}{\gamma_1(0)} \mathbf{e}_i \\ \mathbf{e}_i \end{bmatrix} \in \mathbb{R}^{2n}, \quad i = 1, \dots, n, \quad (32)$$

$$\lambda_2 \implies \mathbf{z}_{n+i} = \begin{bmatrix} -\frac{\sigma_3 - \lambda_2}{\sigma_2} \mathbf{e}_i \\ \mathbf{e}_i \end{bmatrix} = \begin{bmatrix} -\frac{\gamma_1(0)}{\gamma_2(0)} \mathbf{e}_i \\ \mathbf{e}_i \end{bmatrix} \in \mathbb{R}^{2n}, \quad i = 1, \dots, n. \quad (33)$$

The last result implies that in order to satisfy the conditions (18), for any $1 \leq j \neq h \leq P \leq 2n$, at iteration $k = 1$, it suffices to set the initial particle position and velocity (respectively of the i -th and $(n + i)$ -th particle) according with the following initializations

$$\mathbf{X}(0)^{(i)} = \rho_i^1 \mathbf{z}_i, \quad \rho_i^1 \in \mathbb{R} \setminus \{0\}, \quad i = 1, \dots, n \quad (34)$$

and

$$\mathbf{X}(0)^{(n+i)} = \rho_i^2 \mathbf{z}_{n+i}, \quad \rho_i^2 \in \mathbb{R} \setminus \{0\}, \quad i = 1, \dots, n. \quad (35)$$

To sum up, in case the Assumptions 1 and 2 hold (which also implies that no randomness is used in DPSO iteration), then

- when $P \leq 2n$, the choice (34)-(35) of the particles' position and velocity guarantees that *the components of velocity of the free responses* of the particles will be *orthogonal at iteration $k = 1$* ;
- in case $P > 2n$, the user can adopt the initialization (34)-(35) of the particles position and velocity for $2n$ particles, while for the remaining $(P - 2n)$ particles the choice of position and velocity is arbitrary.

We conclude this section by remarking that the choice of the nonzero coefficients $\rho_i^1, \rho_i^2, i = 1, \dots, n$ in (34)-(35) is actually arbitrary, so that they can be suitably set by the user and are problem dependent. Anyway, for our numerical experience we adopted the choice $\rho_i^1 = \rho_i^2 = 0.5$, for $i = 1, \dots, n$, in order to maintain the feasibility of initial particles position (see also Section 7).

5 Possible drawbacks: a simple example

Here we detail reasons for the fact that on several real problems the setting (34)-(35) for the initial DPSO population might be still inadequate. This seems an important preliminary step in order to justify the analysis in the second part of this paper. To evaluate the effectiveness of the initial setting (34)-(35), we first tested it on the solution of a linearly constrained nondifferentiable portfolio selection problem, described below.

We consider a simplified version of the portfolio selection model recently proposed in [16]. The model in [16] uses a coherent risk measure based on the combination of lower and upper moments of different orders of the portfolio return distribution. Such a measure can manage non-Gaussian distributions of asset returns, to reflect different investors' risk attitudes. The simplified model allows *short-selling*, i.e. the amount invested in each asset can be possibly negative. In our model we used the following parameters:

- N : number of possible assets ($N = 20$ in our numerical example);
- r_{ex} : minimum expected return of the portfolio;
- r_i : random variable indicating the return of the i -th asset, for $i = 1, \dots, N$;
- p : index of the norm used in the risk measure of the portfolio, with $p \geq 1$ ($p = 3$ in our numerical example);
- a : parameter of the risk measure, with $0 \leq a \leq 1$ ($a = 0.5$ in our numerical example).

Moreover, the variables in our model are described as follows:

- x_i : (with $x_i \in \mathbb{R}$) quantity of the portfolio invested in the i -th asset, for $i = 1, \dots, N$;
- \mathbf{r} : portfolio return.

In addition, $E[\mathbf{y}]$ indicates the *expected value* of the random argument \mathbf{y} , while \mathbf{y}^- indicates $\max\{0, -\mathbf{y}\}$ and \mathbf{y}^+ indicates $(-\mathbf{y})^-$. Finally, we use the symbol \hat{r}_i for $E[r_i]$. Given the above notation, the expected portfolio return $E[\mathbf{r}]$ is equal to

$$E[\mathbf{r}] = \sum_{i=1}^N \hat{r}_i x_i,$$

and our overall simplified constrained portfolio selection problem is as follows

$$\min_{\mathbf{x}} \rho_{a,p}(\mathbf{r}) = a\|\mathbf{r} - E[\mathbf{r}]\|_1 + (1-a)\|\mathbf{r} - E[\mathbf{r}]\|_p - E[\mathbf{r}] \quad (36)$$

$$s.t. \quad E[\mathbf{r}] \geq r_{ex} \quad (37)$$

$$\sum_{i=1}^N x_i = 1. \quad (38)$$

In this formulation, the risk measure $\rho_{a,p}(\mathbf{r})$ in (36) is coherent (i.e. it satisfies some formal properties which are appealing for investors) as proved by [10]. Observe also that the norm $\|\cdot\|_1$ makes $\rho_{a,p}(\mathbf{r})$ nondifferentiable. The constraint (37) imposes minimum desired expected return of the portfolio, while (38) imposes a budget constraint but does not impede indeed short-selling. As for the numerical instance, we considered daily prices of 20 assets in Italian FTSE MIB stock-exchange index. Finally, details on the computation of $\rho_{a,p}(\mathbf{r})$ using the daily prices can be found in [16].

In order to approximately solving by DPSO the constrained non smooth problem above, borrowing the idea by [16], we considered the following unconstrained reformulation of (36)-(38)

$$\min_{\mathbf{x} \in \mathbb{R}^N} P(\mathbf{x}; \varepsilon), \quad (39)$$

where we used the exact penalty function

$$P(\mathbf{x}; \varepsilon) = \rho_{a,p}(\mathbf{r}) + \frac{1}{\varepsilon} \left[\max \left\{ 0, r_{ex} - \sum_{i=1}^N \hat{r}_i x_i \right\} + \left| \sum_{i=1}^N x_i - 1 \right| \right] \quad (40)$$

and for simplicity we set the parameter $\varepsilon = 10^{-4}$ in our numerical experience (see also [16] for details on exact penalty approaches). Note that the reformulation (39) is non differentiable and admits in general several solutions, so that DPSO was *specifically adopted to provide fast approximate solutions on several scenarios* (unlike the shape design problem in Section 7, here a few seconds of computation on a laptop are allowed). This is indeed a typical application where tradesmen often claim for a quick solution on different scenarios, rather than a unique accurate solution to propose to their customers. In particular, the initialization (34)-(35) was experienced, and in far less than 200 DPSO iterations it provided really effective results in terms of fast minimization of the fitness function $P(\mathbf{x}; \varepsilon)$ and risk measure $\rho_{a,p}(\mathbf{r})$ (see Figures 2 and 3 for a typical numerical instance), along with feasibility of the final solution. However, we also observed that the initialization (34)-(35) tends to provide a sparse solution, which reduces diversification and might be therefore of scarce interest for some investors. In particular, on different scenarios the final solution provided by the initialization (34)-(35) yielded a portfolio including just 2-7 assets (depending on the scenario considered), which is often too restrictive for many investors. This was a consequence of the corresponding *sparsity* (i.e. a few nonzero entries of the vectors) of the initialization (34)-(35). In the next section we discuss a suitable modification to the initialization (34)-(35), which takes into account and possibly fixes the latter drawback.

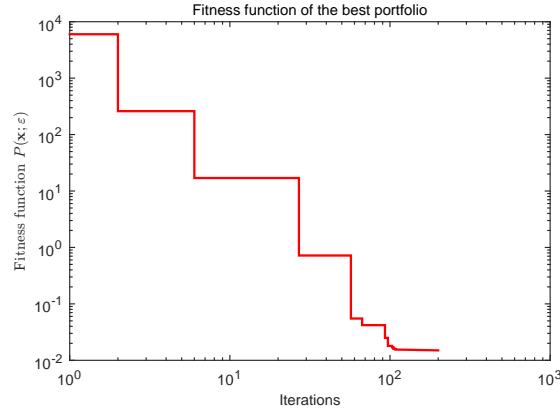


Fig. 2 Fitness function $P(\mathbf{x}; \varepsilon)$ vs. the number of iterations, when the initialization (34)-(35) is experienced.

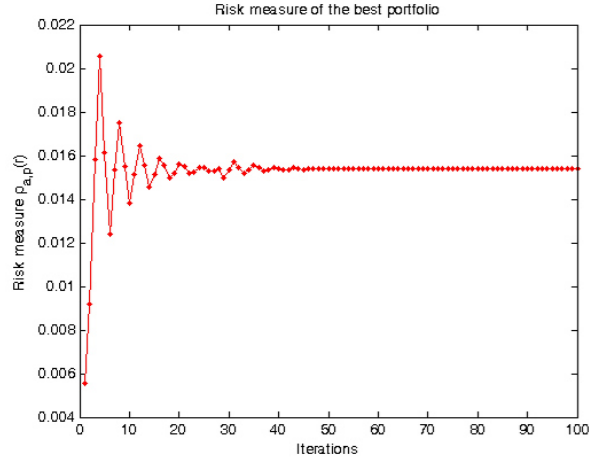


Fig. 3 Risk measure $\rho_{a,p}(\mathbf{r})$ vs. the number of iterations, when the initialization (34)-(35) is experienced.

6 A dense modification

According with the discussion in the last section, here we want to provide a modification of DPSO initialization (34)-(35), with the specific aim to possibly pursue a *dense* final solution for problem (1). We remark that the proposal in this section strongly differs from [20] and the analysis in Section 4.2. Indeed, here we specifically focus on the issue of density (i.e. avoiding a large number of zero entries) for the final solution provided by DPSO. On this purpose, let be given the $2n$ orthonormal eigenvectors \mathbf{z}_i , $i = 1, \dots, 2n$, in (32)-(33) of the symmetric matrix \mathbf{A} in (19). The vectors $\mathbf{z}_1 \dots, \mathbf{z}_{2n}$ are also mutually conjugate with respect to \mathbf{A} , i.e. conditions (31) hold. Now, let without loss of generality the eigenvectors $\mathbf{z}_1 \dots, \mathbf{z}_{2n}$ be associated respectively with the two eigenvalues $\mu_1 = 0$ and $\mu_2 = [\gamma_1(1)]^2 + [\gamma_2(1)]^2 > 0$ of \mathbf{A} , as

$$\begin{aligned} \mathbf{A}\mathbf{z}_i &= \mu_1\mathbf{z}_i, & i &= 1, \dots, n, \\ \mathbf{A}\mathbf{z}_i &= \mu_2\mathbf{z}_i, & i &= n+1, \dots, 2n. \end{aligned} \tag{41}$$

Starting from the motivations suggested in Section 5, in place of vectors $\mathbf{z}_1, \dots, \mathbf{z}_{2n}$ we preliminarily consider the novel set of (*dense*) vectors $\mathbf{w}_1, \dots, \mathbf{w}_{2n}$ as

$$\mathbf{w}_i = \mathbf{z}_i - \alpha \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{z}_j - \gamma \sum_{j=n+1}^{2n} \mathbf{z}_j, \quad i = 1, \dots, n, \quad \alpha, \gamma \in \mathbb{R}, \quad (42)$$

$$\mathbf{w}_t = \mathbf{z}_t - \beta \sum_{\substack{j=n+1 \\ j \neq t}}^{2n} \mathbf{z}_j - \delta \sum_{j=1}^n \mathbf{z}_j, \quad t = n+1, \dots, 2n, \quad \beta, \delta \in \mathbb{R}. \quad (43)$$

Then, we want to compute (if any) the values of α , β , γ and δ such that the following conjugacy conditions hold

$$\mathbf{w}_h^T \mathbf{A} \mathbf{w}_k = 0, \quad \text{for any } 1 \leq h \neq k \leq 2n.$$

Finally, we will use $\mathbf{w}_1, \dots, \mathbf{w}_{2n}$ in (42)-(43) in place of $\mathbf{z}_1, \dots, \mathbf{z}_{2n}$.

From (31), (41) and (42), and setting $1 \leq h \neq k \leq n$ we can impose

$$\begin{aligned} 0 &= \mathbf{w}_h^T \mathbf{A} \mathbf{w}_k = \left[\mathbf{z}_h - \alpha \sum_{\substack{j=1 \\ j \neq h}}^n \mathbf{z}_j - \gamma \sum_{j=n+1}^{2n} \mathbf{z}_j \right]^T \mathbf{A} \left[\mathbf{z}_k - \alpha \sum_{\substack{j=1 \\ j \neq k}}^n \mathbf{z}_j - \gamma \sum_{j=n+1}^{2n} \mathbf{z}_j \right] \\ &= -\alpha \mathbf{z}_h^T \mathbf{A} \mathbf{z}_h - \alpha \mathbf{z}_k^T \mathbf{A} \mathbf{z}_k + \alpha^2 \sum_{\substack{j=1 \\ j \neq h, k}}^n \mathbf{z}_j^T \mathbf{A} \mathbf{z}_j + \gamma^2 \sum_{j=n+1}^{2n} \mathbf{z}_j^T \mathbf{A} \mathbf{z}_j \\ &= -\alpha \mu_1 - \alpha \mu_1 + \alpha^2 (n-2) \mu_1 + \gamma^2 n \mu_2, \end{aligned}$$

and recalling that $\mu_1 = 0$ the latter relations are satisfied if and only if

$$\forall \alpha \in \mathbb{R}, \quad \gamma = 0. \quad (44)$$

Similarly, by (31), (41) and (43), and setting now $n+1 \leq h \neq k \leq 2n$, we can impose

$$\begin{aligned} 0 &= \mathbf{w}_h^T \mathbf{A} \mathbf{w}_k = \left[\mathbf{z}_h - \beta \sum_{\substack{j=n+1 \\ j \neq h}}^{2n} \mathbf{z}_j - \delta \sum_{j=1}^n \mathbf{z}_j \right]^T \mathbf{A} \left[\mathbf{z}_k - \beta \sum_{\substack{j=n+1 \\ j \neq k}}^{2n} \mathbf{z}_j - \delta \sum_{j=1}^n \mathbf{z}_j \right] \\ &= -\beta \mu_2 - \beta \mu_2 + \beta^2 (n-2) \mu_2 + \delta^2 n \mu_1, \end{aligned}$$

and recalling that $\mu_2 > 0$ the latter relations are satisfied for $n > 2$ if and only if

$$\beta \in \left\{ 0, \frac{2}{n-2} \right\}, \quad \forall \delta \in \mathbb{R}. \quad (45)$$

Now, taken the vectors \mathbf{w}_i , for $i \in \{1, \dots, n\}$ in (42), and \mathbf{w}_t , for $t \in \{n+1, \dots, 2n\}$ in (43) we have by (31), (44) and (45)

$$\begin{aligned} \mathbf{w}_i^T \mathbf{A} \mathbf{w}_t &= \left[\mathbf{z}_i - \alpha \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{z}_j \right]^T \mathbf{A} \left[\mathbf{z}_t - \frac{2}{n-2} \sum_{\substack{j=n+1 \\ j \neq t}}^{2n} \mathbf{z}_j - \delta \sum_{j=1}^n \mathbf{z}_j \right] \\ &= -\delta \mathbf{z}_i^T \mathbf{A} \mathbf{z}_i + \alpha \delta \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{z}_j^T \mathbf{A} \mathbf{z}_j = -\delta \mu_1 + (n-1) \alpha \delta \mu_1 = 0. \end{aligned}$$

Thus, taken the orthonormal eigenvectors $\mathbf{z}_1, \dots, \mathbf{z}_{2n}$ of \mathbf{A} in (32)-(33), along with the values

$$\alpha \in \mathbb{R} \setminus \{0\}, \quad \beta = \frac{2}{n-2}, \quad \gamma = 0, \quad \delta \in \mathbb{R} \setminus \{0\}, \quad (46)$$

then the vectors $\mathbf{w}_1, \dots, \mathbf{w}_n, \mathbf{w}_{n+1}, \dots, \mathbf{w}_{2n}$ in (42)-(43)

1. are mutually \mathbf{A} -conjugate,
2. are *dense*, i.e. they contain a large number of non zero entries.

Now we are interested about finding the values ε_i , $i = 1, \dots, 2n$ such that

$$\|\varepsilon_i \mathbf{w}_i\| = 1, \quad i = 1, \dots, 2n,$$

i.e. $\{\hat{\mathbf{w}}_i\}$ (with $\hat{\mathbf{w}}_i = \varepsilon_i \mathbf{w}_i$) provides a set of $2n$ *unit* mutually \mathbf{A} -conjugate directions $\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_{2n}$, using the vectors $\mathbf{w}_1, \dots, \mathbf{w}_{2n}$. To this purpose we distinguish between the cases $i \in \{1, \dots, n\}$ and $i \in \{n+1, \dots, 2n\}$. In case $i \in \{1, \dots, n\}$ we have by (31)

$$\begin{aligned} 1 &= \|\varepsilon_i \mathbf{w}_i\| = \varepsilon_i \left\{ \left[\begin{array}{c} \mathbf{z}_i - \alpha \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{z}_j \end{array} \right]^T \left[\begin{array}{c} \mathbf{z}_i - \alpha \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{z}_j \end{array} \right] \right\}^{1/2} \\ &= \varepsilon_i \left[\|\mathbf{z}_i\|^2 + \alpha^2 \sum_{\substack{j=1 \\ j \neq i}}^n \|\mathbf{z}_j\|^2 \right]^{1/2} = \varepsilon_i [1 + \alpha^2(n-1)]^{1/2}, \end{aligned}$$

which is satisfied by choosing

$$\varepsilon_i = \frac{1}{[1 + \alpha^2(n-1)]^{1/2}}.$$

Similarly, in case $i \in \{n+1, \dots, 2n\}$ we have by (31)

$$\begin{aligned} 1 &= \|\varepsilon_i \mathbf{w}_i\| = \varepsilon_i \left\{ \left[\begin{array}{c} \mathbf{z}_i - \frac{2}{n-2} \sum_{\substack{j=n+1 \\ j \neq i}}^{2n} \mathbf{z}_j - \delta \sum_{j=1}^n \mathbf{z}_j \end{array} \right]^T \left[\begin{array}{c} \mathbf{z}_i - \frac{2}{n-2} \sum_{\substack{j=n+1 \\ j \neq i}}^{2n} \mathbf{z}_j - \delta \sum_{j=1}^n \mathbf{z}_j \end{array} \right] \right\}^{1/2} \\ &= \varepsilon_i \left[\|\mathbf{z}_i\|^2 + \frac{4}{(n-2)^2}(n-1) + \delta^2 n \right] = \varepsilon_i \left[1 + \frac{4}{(n-2)^2}(n-1) + n\delta^2 \right] \\ &= \varepsilon_i \frac{n^2 + \delta^2 n}{(n-2)^2}, \end{aligned}$$

which is satisfied by choosing

$$\varepsilon_i = \frac{(n-2)^2}{n^2 + \delta^2 n}.$$

To sum up, using (32)-(33) we explicitly report the following vectors $\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_{2n}$ (i.e. the vectors in (42)-(43) after normalization),

$$\hat{\mathbf{w}}_i = \frac{1}{[1 + \alpha^2(n-1)]^{1/2}} \left[\mathbf{z}_i - \alpha \sum_{\substack{j=1 \\ j \neq i}}^n \mathbf{z}_j - \gamma \sum_{j=n+1}^{2n} \mathbf{z}_j \right] \quad (47)$$

$$= \frac{1}{[1 + \alpha^2(n-1)]^{1/2}} \begin{bmatrix} -\alpha \frac{\gamma_2(0)}{\gamma_1(0)} + \gamma \frac{\gamma_1(0)}{\gamma_2(0)} \\ \vdots \\ -\alpha \frac{\gamma_2(0)}{\gamma_1(0)} + \gamma \frac{\gamma_1(0)}{\gamma_2(0)} \\ \frac{\gamma_2(0)}{\gamma_1(0)} + \gamma \frac{\gamma_1(0)}{\gamma_2(0)} \leftarrow \boxed{i} \\ -\alpha \frac{\gamma_2(0)}{\gamma_1(0)} + \gamma \frac{\gamma_1(0)}{\gamma_2(0)} \\ \vdots \\ -\alpha \frac{\gamma_2(0)}{\gamma_1(0)} + \gamma \frac{\gamma_1(0)}{\gamma_2(0)} \\ -\alpha - \gamma \\ \vdots \\ -\alpha - \gamma \\ 1 - \gamma \leftarrow \boxed{n+i} \\ -\alpha - \gamma \\ \vdots \\ -\alpha - \gamma \end{bmatrix}, \quad i = 1, \dots, n, \quad (48)$$

$$\hat{\mathbf{w}}_t = \frac{(n-2)^2}{n^2 + \delta^2 n} \left[\mathbf{z}_t - \beta \sum_{\substack{j=n+1 \\ j \neq t}}^{2n} \mathbf{z}_j - \delta \sum_{j=1}^n \mathbf{z}_j \right] \quad (49)$$

$$= \frac{(n-2)^2}{n^2 + \delta^2 n} \begin{bmatrix} -\delta \frac{\gamma_2(0)}{\gamma_1(0)} + \beta \frac{\gamma_1(0)}{\gamma_2(0)} \\ \vdots \\ -\delta \frac{\gamma_2(0)}{\gamma_1(0)} + \beta \frac{\gamma_1(0)}{\gamma_2(0)} \\ -\delta \frac{\gamma_2(0)}{\gamma_1(0)} - \frac{\gamma_1(0)}{\gamma_2(0)} \leftarrow \boxed{t-n} \\ -\delta \frac{\gamma_2(0)}{\gamma_1(0)} + \beta \frac{\gamma_1(0)}{\gamma_2(0)} \\ \vdots \\ -\delta \frac{\gamma_2(0)}{\gamma_1(0)} + \beta \frac{\gamma_1(0)}{\gamma_2(0)} \\ -\delta - \beta \\ \vdots \\ -\delta - \beta \\ -\delta + 1 \leftarrow \boxed{t} \\ -\delta - \beta \\ \vdots \\ -\delta - \beta \end{bmatrix}, \quad t = n+1, \dots, 2n, \quad (50)$$

The vectors in (48)-(50) represent *our final proposal of DPSO initialization for unconstrained optimization*. They have unit norm and are \mathbf{A} -conjugate, with the choice (46) of the coefficients α , β , γ and δ . Now, we urge to prove that $\mathbf{w}_1, \dots, \mathbf{w}_{2n}$ (and consequently $\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_{2n}$) are uniformly *linearly independent*, i.e. roughly speaking the angles among them remain sufficiently bounded away from zero. This fact is of great relevance in the light of scattering the particles in the search space, and it is not immediately evident, since the matrix \mathbf{A} is *only* positive semidefinite.

On this purpose, for the sake of simplicity we prove below that the quantity $\det(\mathbf{w}_1 \dotscots \mathbf{w}_{2n})$ is sufficiently (and uniformly with respect to n) bounded away from zero, which also implies that $\det(\hat{\mathbf{w}}_1 \dotscots \hat{\mathbf{w}}_{2n})$ is sufficiently bounded away from zero.

Proposition 3 *Given the vectors $\mathbf{w}_1, \dots, \mathbf{w}_{2n}$ in (42)-(43), when $n \neq 2$ then*

$$\det(\mathbf{w}_1 \dotscots \mathbf{w}_{2n}) = (1 + \alpha)^{n-1} [1 - (n-1)\alpha] \frac{(3n-4)n^{n-1}}{(n-2)^n}. \quad (51)$$

Proof

For the sake of clarity the proof is in the Appendix. \diamond

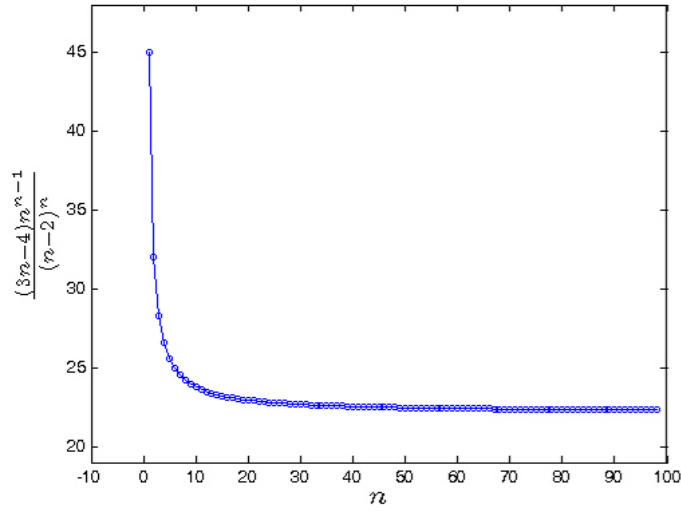


Fig. 4 Plot of the quantity $\frac{(3n-4)n^{n-1}}{(n-2)^n}$ (which is independent of α) versus n , as reported in (51). Note that this quantity is uniformly bounded away from zero, being $\lim_{n \rightarrow \infty} (3n-4)n^{n-1}/(n-2)^n = 3/e^{-2} \approx 22.17$.

In Figure 4 we show a plot of the quantity $(3n-4)n^{n-1}/(n-2)^n$ in (51) (which is independent of α), versus $n \in \{3, \dots, 100\}$, showing that for α satisfying

$$\alpha \notin [(-1 - \sigma), (-1 + \sigma)] \cup \left[\left(\frac{1}{n-1} - \sigma \right), \left(\frac{1}{n-1} + \sigma \right) \right], \quad \sigma > 0,$$

the quantity $\det(\mathbf{w}_1 \dotscots \mathbf{w}_{2n})$ is indeed uniformly bounded away from zero in this interval. Observe that the last limitation on α is definitely a non-restrictive assumption.

As regards the results on the portfolio selection problem analyzed in Section 5, we compared the initialization (48)-(50) (using the parameters $\alpha = 0.25$, $\beta = 2/(n-2)$, $\gamma = 0$ and $\delta = 0.75$) vs. the initialization (34)-(35), obtaining analogous results in terms of decrease of the fitness function $P(\mathbf{x}; \varepsilon)$ and risk measure $\rho_{a,p}(\mathbf{r})$ (i.e. pictures very similar to Figures 2 and 3 are obtained, with a slight preference for (34)-(35) with respect to (48)-(50)). However, adopting (48)-(50) we drastically improved the density of the final approximate solution, inasmuch as now almost all the assets were included in the portfolio. The latter result should not be surprising, since in (34)-(35) the initialization of any particle contains just two non zero entries, while in (48)-(50) all the entries of the vectors $\hat{\mathbf{w}}_i$ and $\hat{\mathbf{w}}_t$ are nonzero.

Of course, we have experienced different settings for the parameters α , β , γ and δ , along with the penalty parameter ε in (40), obtaining similar results. In this regard, a change of settings can

cause a different final solution obtained, as expected considering that in general there are different minimizers of (40).

7 Examples on analytic benchmark and ship design problems

Simulation-based design optimization (SBDO) paradigm supports the design process of complex engineering systems and recently replaced the traditional expensive build and test approach. SBDO integrates three key elements: computer simulations, design modification methods and optimization algorithms. Within the context of ship/ocean applications global derivative-free algorithms are widely used, since objective and constraint functions - often provided by black box tools - are non convex and noisy, their derivatives are not directly provided, and several different local minima cannot be excluded *a priori*. Moreover, the concurrent use of global techniques and CPU-time expensive solvers makes the optimization process computationally expensive, and still represents an algorithmic and technological challenge. It should be noted that several derivative-free global optimization algorithms in the literature are probabilistic and use random coefficients to enhance the dynamics of the swarm in the optimization process. Therefore, statistically significant results can be derived only through extensive numerical campaigns. Such an approach is often too expensive in SBDO for industrial applications, especially when high-fidelity physics-based solvers are used as analysis tools. For these reasons deterministic approaches, such as DPSO, have been developed and successfully applied to SBDO, including hydrodynamic problems, providing comparisons among local methods [7] and/or random PSO [9,38].

Preliminarily to approach real-world ship design optimization problems, a parametric analysis has been conducted here on an analytic benchmark, including twelve test functions. This small test set both contains continuous and discontinuous, differentiable and non-differentiable, separable and non-separable, unimodal and multimodal test functions, which are used in order to assess the performances of the novel initialization proposed. Specifically, four particles initializations, in terms of initial position and velocity, have been compared corresponding to:

1. Hammersly sequence sampling (HSS, [50]) distribution on domain and bounds with non-null velocity for problems with dimensionality $n < 10$ and distribution on domain with non-null velocity for problems with dimensionality $n \geq 10$, defined as the most promising initialization in the study provided by [42];
2. ORTHOint initialization, as suggested by [20];
3. ORTHOint+, the novel initialization suggested herein ($\alpha = 0.25$, $\beta = 2/(6 - 2) = 0.5$, $\gamma = 0$ and $\delta = 0.75$), using $2n + 2n$ particles, being the first $2n$ particles initialized as in (47)–(50), while the last $2n$ particles are initialized as in (47)–(50) with a sign inversion;
4. ORTHOint#, a combination of ORTHOint and ORTHOint+ ($\alpha = 0.25$, $\beta = 2/(6 - 2) = 0.5$, $\gamma = 0$ and $\delta = 0.75$), using $2n + 2n$ particles (the first $2n$ particles being initialized using ORTHOint and the last $2n$ particles initialized as in ORTHOint+).

The most promising DPSO implementation identified in the study by [42] has been used for our numerical experience, which is characterized by: number of particles (P) equal to 4 times the number of design variables (n), set of coefficient defined by [13] (*i.e.*, $\chi = 0.721$, $c_1 = c_2 = 1.655$), and a semi-elastic wall-type approach for box constraints, in place of the exact penalty approach considered in Section 5. The algorithm showing the semi-elastic wall-type approach is presented in Algorithm 1 and in Figure 5, for further details the reader can refer to [42].

Subsequently, the same DPSO and initialization sets have been applied to the ship design optimization problem.

Regardless of our theoretical analysis, we observe that proposing a number of particles proportional to the scale n of the problem (*i.e.* $P = 2n$ or $P = 2n + 2n$) might appear unusual; however, it is suggested by a couple of additional considerations. First, note that a similar result also holds for Direct Search methods (see Section 1), and it is a crucial prerequisite in order to prove their

global convergence properties. Second, design problems are quite often numerically expensive, so that the computation of even coarse approximate solutions may require a relatively large number of function evaluations. This is a serious drawback which suggested to preferably focus in this paper on the efficiency of the overall method, as measured by the final number of function evaluations, rather than on the number of PSO particles introduced. Surely the latter statement might be questionable in case other applications were considered, where the number of particles can play a more relevant role (see e.g. [31]).

We want to show that the initializations ORTHOinit+ and ORTHOinit#, which take advantage of the theory developed in the current paper, show a preferable performance on the proposed test set.

Algorithm 1 Semi-elastic wall-type approach (SEW)

```

1: for  $j = 1, P$  do
2:   for  $i = 1, n$  do
3:     if  $(x_{i,j}^k > u_i)$  OR  $(x_{i,j}^k < l_i)$  then
4:        $v_{i,j}^k = -v_{i,j}^k / [\chi(c_1 + c_2)]$ 
5:     end if
6:   end for
7: end for
  
```

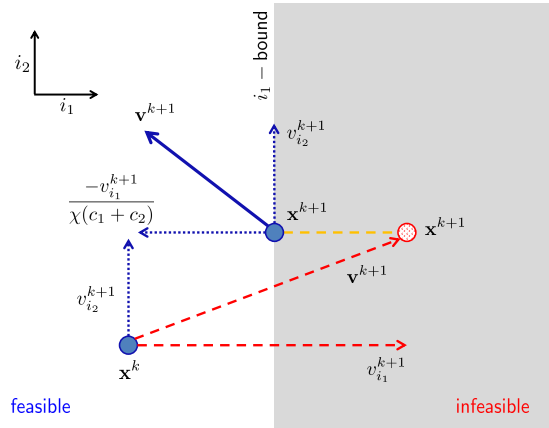


Fig. 5 Semi-elastic wall-type approach applied in the transition from k -th to $(k + 1)$ -th PSO iteration [42].

7.1 Analytic test problems

The set of analytic functions (see Table 1), which represents a subset of the ones used by [29,30], has been recently used by [38], to investigate the performances of random PSO and DPSO, as formulated by [42].

In order to possibly avoid the introduction of any additional bias in our numerical experience, following the approach in [42] the next three absolute performance criteria have been used to assess the initialization performances

$$\Delta_x = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{x_{i,\min} - x_{i,\min}^*}{R_i} \right)^2}; \quad \Delta_f = \frac{f_{\min} - f_{\min}^*}{f_{\max}^* - f_{\min}^*}; \quad \Delta_t = \sqrt{\frac{\Delta_x^2 + \Delta_f^2}{2}}. \quad (52)$$

Here Δ_x represents the normalized Euclidean distance between the minimum position found by the algorithm \mathbf{x}_{\min} and the analytic minimum position \mathbf{x}_{\min}^* , and $R_i = |u_i - l_i|$ is the range of the i -th variable. Δ_f is the normalized distance in the image space where f_{\min} is the minimum found

Table 1 Analytic test functions.

Name	$f(\mathbf{x})$	Bounds $1 \leq \mathbf{x} \leq \mathbf{u}$	Opt. f_{\min}^*
Ackley	$f_1(\mathbf{x}) = 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} + 20 + e$	$-5.00 \leq x_i \leq 4.00$	0.00
Alpine	$f_2(\mathbf{x}) = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	$-9.00 \leq x_i \leq 7.00$	0.00
Dixon-Price	$f_3(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^n [i(2x_i^2 - x_{i-1})^2]$	$-10.0 \leq x_i \leq 10.0$	0.00
Griewank	$f_4(\mathbf{x}) = 1 + \sum_{i=1}^n \frac{x_i}{4000} - \prod_{i=1}^n \cos(x_i/\sqrt{i})$	$-100. \leq x_i \leq 90.0$	0.00
Levy n.5	$f_5(\mathbf{x}) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} [(y_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1}))] + (y_n - 1)^2 \right\}$ with $y_i = 1 + \frac{1}{4}(x_i - 1)$	$-10.0 \leq x_i \leq 10.0$	0.00
Mishra n.11	$f_6(\mathbf{x}) = \left[\frac{1}{n} \sum_{i=1}^n x_i - \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}} \right]^2$	$-10.0 \leq x_i \leq 9.00$	0.00
Rastrigin	$f_7(\mathbf{x}) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	$-5.12 \leq x_i \leq 4.12$	0.00
Rosenbrock	$f_8(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	$-5.00 \leq x_i \leq 10.0$	0.00
Sphere	$f_9(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$-5.00 \leq x_i \leq 4.00$	0.00
Styblinski-Tang	$f_{10}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i) + 39.2n$	$-5.00 \leq x_i \leq 5.00$	0.00
Trigonometric n.2	$f_{11}(\mathbf{x}) = \sum_{i=1}^n 8 \sin^2 [7(x_i - 0.9)^2] + 6 \sin^2 [14(x_i - 0.9)^2] + (x_i - 0.9)^2$	$-500. \leq x_i \leq 500.$	0.00
Zacharov	$f_{12}(\mathbf{x}) = \sum_{i=1}^n x_i^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i \right)^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i \right)^4$	$-5.00 \leq x_i \leq 10.0$	0.00

by the algorithm, f_{\min}^* is the analytic minimum, and f_{\max}^* the analytic maximum of the function $f(\mathbf{x})$ in the domain. Δ_t is the combination of Δ_x and Δ_f and is used for the overall performances assessment.

Analytic test problems have been studied with dimensionality $n = 6$ (the same dimension of the real-world ship design problem) and $n = 50$, using a total number of function evaluations (N_{feval}) equal to 2,400.

Table 2 shows the best performing initialization for each function and on average (on the twelve analytic test functions), with respect to the three metrics, respectively for the problem with dimensionality $n = 6$ and $n = 50$. The averaged results summarized for $n = 6$ indicate that the novel initialization ORTHOinit+ outperforms the others, both referring to the variable space (Δ_x) and to the function space (Δ_f). It might also be noted that ORTHOinit# initialization shows good performances in the variable space, whereas HSS is less efficient, as observed in previous studies [20]. Referring to Δ_t , ORTHOinit+ shows the best performance overall, followed by ORTHOinit#. The averaged results for $n = 50$ indicate that ORTHOinit# initialization outperforms the others, both referring to the variable space (Δ_x) and to the function space (Δ_f). It might also be noted that ORTHOinit+ initialization shows good performances both in variable and function space, whereas HSS is less efficient initialization, as observed in previous studies [20]. Referring to Δ_t , ORTHOinit# shows the best performance overall, followed by ORTHOinit+.

Figure 6 shows DPSO convergence history for each test function with dimensionality $n = 50$, conditional to the four initializations.

Table 2 Analytic test functions: average performances results.

$f(\mathbf{x})$	Initialization	$n = 6$			$n = 50$		
		Δ_x	Δ_f	Δ_t	Δ_x	Δ_f	Δ_t
$f_1(\mathbf{x})$	HSS	8.869E-02	2.220E-01	1.690E-01	7.673E-02	2.692E-01	1.980E-01
	ORTHOinit	9.786E-02	2.414E-01	1.842E-01	1.065E-01	2.630E-01	2.006E-01
	ORTHOinit+	7.777E-09	7.010E-08	4.987E-08	2.192E-03	7.358E-03	5.429E-03
	ORTHOinit#	9.785E-02	2.414E-01	1.842E-01	1.729E-05	4.633E-05	3.497E-05
$f_2(\mathbf{x})$	HSS	2.195E-01	0.000E-00	1.552E-01	4.605E-01	9.281E-03	3.257E-01
	ORTHOinit	9.818E-02	0.000E-00	6.942E-02	9.816E-02	0.000E-00	6.941E-02
	ORTHOinit+	9.817E-02	0.000E-00	6.942E-02	3.103E-02	1.250E-02	2.366E-02
	ORTHOinit#	9.816E-02	0.000E-00	6.941E-02	9.818E-02	0.000E-00	6.942E-02
$f_3(\mathbf{x})$	HSS	2.837E-02	7.558E-07	2.006E-02	6.963E-02	4.460E-04	4.923E-02
	ORTHOinit	2.978E-02	7.558E-07	2.106E-02	2.563E-02	1.186E-08	1.812E-02
	ORTHOinit+	1.061E-02	2.710E-07	7.499E-03	2.540E-02	1.355E-08	1.796E-02
	ORTHOinit#	2.733E-03	6.177E-08	1.933E-03	2.563E-02	1.186E-08	4.924E-02
$f_4(\mathbf{x})$	HSS	6.193E-02	1.300E-02	4.475E-02	3.823E-02	1.317E-02	2.859E-02
	ORTHOinit	3.688E-02	1.565E-02	2.833E-02	3.543E-02	4.731E-03	2.527E-02
	ORTHOinit+	1.677E-10	9.518E-20	1.186E-10	4.945E-02	9.264E-03	3.568E-02
	ORTHOinit#	3.042E-02	2.215E-02	2.661E-02	4.128E-06	1.236E-08	2.919E-06
$f_5(\mathbf{x})$	HSS	1.736E-03	1.839E-06	1.227E-03	7.324E-02	5.080E-03	5.191E-02
	ORTHOinit	8.539E-07	3.922E-13	6.038E-07	4.348E-02	4.565E-03	3.092E-02
	ORTHOinit+	0.000E-00	2.646E-16	1.871E-16	6.638E-02	2.066E-03	4.696E-02
	ORTHOinit#	1.989E-06	4.150E-12	1.407E-06	1.056E-02	5.609E-05	5.191E-02
$f_6(\mathbf{x})$	HSS	3.654E-02	8.427E-04	2.585E-02	6.517E-02	7.136E-03	4.636E-02
	ORTHOinit	1.822E-04	6.919E-08	1.288E-04	5.969E-05	5.109E-10	4.221E-05
	ORTHOinit+	2.216E-09	1.772E-17	1.567E-09	2.082E-02	4.334E-05	1.472E-02
	ORTHOinit#	3.768E-09	3.817E-17	2.665E-09	8.286E-08	1.861E-15	5.859E-08
$f_7(\mathbf{x})$	HSS	1.296E-01	5.063E-02	9.840E-02	1.400E-01	1.544E-01	1.833E-01
	ORTHOinit	9.836E-02	2.599E-02	7.194E-02	1.077E-01	3.429E-02	7.995E-02
	ORTHOinit+	1.088E-01	3.056E-02	7.915E-02	1.752E-02	6.597E-03	1.324E-02
	ORTHOinit#	2.154E-01	1.222E-01	1.751E-01	6.809E-02	1.371E-02	4.911E-02
$f_8(\mathbf{x})$	HSS	6.454E-02	1.373E-06	4.564E-02	6.806E-02	3.971E-05	4.813E-02
	ORTHOinit	5.448E-03	3.410E-09	3.852E-03	6.527E-02	3.641E-06	4.615E-02
	ORTHOinit+	3.974E-09	5.172E-19	2.810E-09	6.554E-02	3.661E-06	4.634E-02
	ORTHOinit#	0.000E-00	0.000E-00	0.000E-00	1.241E-03	4.248E-08	8.779E-04
$f_9(\mathbf{x})$	HSS	2.588E-03	2.170E-05	1.830E-03	4.767E-02	7.362E-03	3.411E-02
	ORTHOinit	2.218E-07	1.594E-13	1.568E-07	1.352E-05	5.919E-10	9.557E-06
	ORTHOinit+	2.653E-13	2.280E-25	1.876E-13	6.836E-06	1.514E-10	4.833E-06
	ORTHOinit#	2.114E-10	1.447E-19	1.495E-10	2.595E-07	2.182E-13	1.835E-07
$f_{10}(\mathbf{x})$	HSS	1.979E-04	4.968E-07	1.399E-04	1.778E-01	1.054E-02	1.259E-01
	ORTHOinit	2.189E-01	1.278E-01	1.793E-01	2.827E-01	2.625E-01	2.728E-01
	ORTHOinit+	1.101E-05	0.000E-00	7.841E-06	2.841E-04	4.195E-07	2.009E-04
	ORTHOinit#	2.146E-05	0.000E-00	1.520E-05	8.017E-02	2.024E-03	5.671E-02
$f_{11}(\mathbf{x})$	HSS	4.254E-03	8.109E-05	3.009E-03	4.115E-02	6.772E-03	2.949E-02
	ORTHOinit	2.738E-04	2.993E-07	1.936E-04	8.560E-04	2.721E-05	6.056E-04
	ORTHOinit+	1.192E-10	5.664E-20	8.429E-11	8.513E-03	2.919E-04	6.023E-03
	ORTHOinit#	0.000E-00	0.000E-00	0.000E-00	2.612E-04	9.182E-07	1.847E-04
$f_{12}(\mathbf{x})$	HSS	5.132E-04	7.905E-12	3.629E-04	1.211E-01	1.194E-12	8.566E-02
	ORTHOinit	1.961E-03	1.135E-10	1.387E-03	1.535E-02	1.917E-14	1.085E-02
	ORTHOinit+	1.705E-09	8.870E-22	1.206E-09	8.932E-03	6.496E-15	6.316E-03
	ORTHOinit#	1.445E-07	4.208E-19	1.022E-07	7.305E-03	4.341E-15	5.165E-03
Average	HSS	5.321E-02	2.388E-02	4.713E-02	1.15E-01	4.029E-02	9.754E-02
	ORTHOinit	4.899E-02	3.424E-02	4.665E-02	6.510E-02	4.743E-02	6.290E-02
	ORTHOinit+	1.804E-02	2.547E-03	1.301E-02	2.467E-02	3.177E-03	1.804E-02
	ORTHOinit#	3.705E-02	3.215E-02	3.811E-02	2.429E-02	1.320E-03	1.726E-02

Once more, no statistic analysis was required for the results presented. Indeed, no random parameters have been included in the four DPSO-based schemes implemented here, inasmuch as the paper focuses on the performance of a deterministic PSO initialization (see also comments in the previous section).

7.2 Engineering problem: ship design optimization

In this subsection we present a real-world ship design optimization problem, in order to steer the work of engineers. It should be noted that design problems often do not have a unique solution,

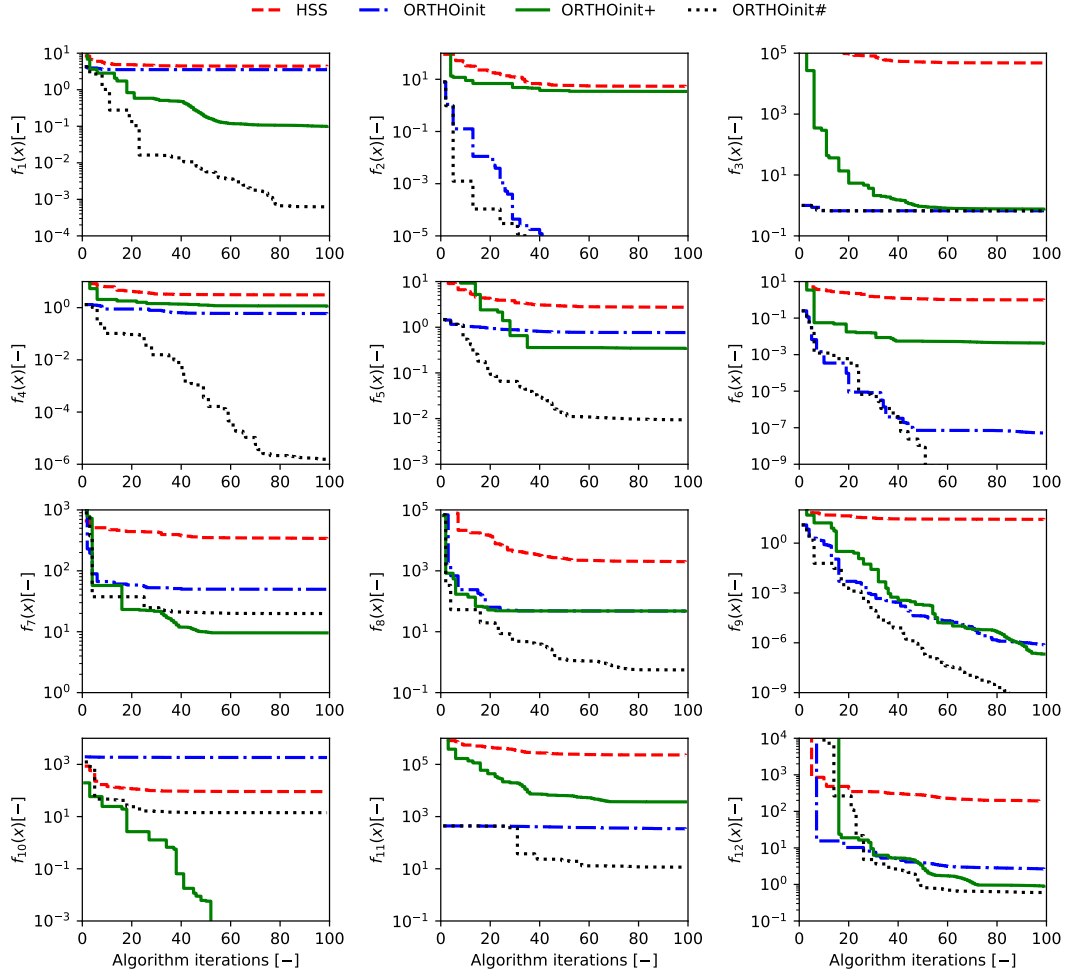


Fig. 6 DPSO convergences conditional to the different initializations ($n = 50$).

due to the intrinsic nature of the problems in hand. Thus, decision makers often prefer to have a number of possible equivalent solutions for these problems, which possibly differ in the values of design variables, in order to choose on the basis of exogenous considerations. Specifically, the SBDO application addresses the hull-form optimization aimed at reducing the total resistance of an USS Arleigh Burke-class destroyer ship, namely the DTMB 5415 model, an early and open-to-public version of the DDG-51, widely used for both experimental [43] and numerical [8] investigations. Figure 7 shows the 5.720 m length replica of the DTMB 5415 model, used for towing tank experiments, as seen at CNR-INM [43]. The DTMB 5415 has been also used as a benchmark in NATO STO Task Groups AVT-204 *Assess the Ability to Optimize Hull Forms of Sea Vehicles for Best Performance in a Sea Environment* [24], AVT-252 *Stochastic Design Optimization for Naval and Aero Military Vehicles* [40]. Moreover, it was used as a test case in variable-accuracy multi-disciplinary design optimization studies, coupling the hydrodynamic analysis with the rigid body equation of motion through multi-disciplinary analysis [28]. Recently, the same problem has been investigated by [19] addressing the dimensionality reduction of the design space, based on Karhunen-Loève expansion technique. For the current application the optimization has been performed on a reduced six-dimensional design space, retaining up to 92% of the original geometric variability, as shown by [19]. Since the purpose of this work is to confirm the theoretical achievements described in the previous sections, a linear potential flow solver developed at CNR-INM [3], namely WARP (WAVE Resistance Program), has been used to evaluate the hydrodynamic performances. Specifically, the

Table 3 DTMB 5415: main particulars and test conditions.

Description	Symbol	Unit	Full scale	Model scale
Displacement	∇	tonnes	8,636	0.549
Length between perpendiculars	L_{pp}	m	142.0	5.720
Beam	B	m	18.90	0.760
Draft	T	m	6.160	0.248
Longitudinal center of gravity	LCG	m	71.60	2.884
Vertical center of gravity	VCG	m	1.390	0.056
Froude Number	Fr	-	0.25	0.25
Reynolds number	Re	-	1.215E+09	9.824E+06

wave resistance computations - based on the double model linear potential flow theory [17] - has been evaluated with pressure integral, whereas the frictional resistance has been estimated using a flat-plate approximation, based on the local Reynolds number [37].

The single-objective shape design optimization problem for DTMB 5415 has been formulated as

$$\begin{aligned}
 & \min_{\mathbf{x} \in \mathbb{R}^n} f = R_T(\mathbf{x}) \\
 & s.t. \quad L_{pp}(\mathbf{x}) = L_{pp,0} \\
 & \quad \quad \nabla(\mathbf{x}) = \nabla_0 \\
 & \quad \quad |\Delta B(\mathbf{x})| \leq 0.05B_0 \\
 & \quad \quad |\Delta T(\mathbf{x})| \leq 0.05T_0 \\
 & \quad \quad V(x) \geq V_0
 \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^6$ is the design variable vector, $R_T : \mathbb{R}^6 \rightarrow \mathbb{R}$ is the total resistance in calm water at 18kn (corresponding to Froude number $Fr = 0.25$), L_{pp} is the length between perpendiculars, ∇ is the displacement, B is the beam, T is the draft, and V is the volume reserved for the sonar in the dome. Subscript ‘0’ indicates parent hull values. Following the approach used by [41], the first two constraints have been automatically satisfied by the geometric scaling, whereas constraints on B and T have been handled using a penalty function method, since the relationship between B/T variations and design variables was not explicitly provided by the orthogonal expansion and geometric scaling. The main particulars of the DTMB 5415 model and the test conditions considered for its optimal design are summarized in Table 3.

The simulations have been performed for the right demi-hull, taking advantage of symmetry about the xz -plane. The computational domain for the free surface has been defined within $1 L_{pp}$ upstream, $3 L_{pp}$ downstream and $1.5 L_{pp}$ sideways. The associated hull grid (90×25 nodes) used (see Figure 8) guarantees solution convergence. The validation of the potential flow analysis performed by WARP for the original hull versus experimental data collected at CNR-INM [33] has been shown by [41]. The optimization problem has been solved using the DPSO schemes presented in Section 7 using again a total number of 2,400 function evaluations. Figures 9 (a) and (b) report the corresponding performances and final optimal design variables for the four initializations. The four DPSO initialization variants have achieved the same objective function reduction (Δf), as shown in Figure 9a, showing equivalent effectiveness. Specifically, ORTHOinit initialization has been the most efficient, as it exactly relies on orthogonal vectors in order to steer the particles. In

**Fig. 7** A 5.720m length model of the DTMB 5415 (CNR-INM model 2340).

other words, using the language of Sections 2, 3 and 4, in ORTHOinit the components of velocity in the free responses associated to the particles are exactly *orthogonal* at step $k = 1$.

On the other hand, ORTHOinit+ is not endowed with the latter orthogonality property, inasmuch as it requires the transformation indicated in (42)-(43), which guarantees uniform linear independence of vectors in place of orthogonality. Thus, ORTHOinit+ seemed less efficient, but thanks to the dense initial positions (as detailed in Section 6) it was able to provide noteworthy different values of the 6 unknowns with respect to ORTHOinit (see the 2nd and 3rd unknown in Figure 9b), which is of great importance for ship designers. Indeed, the capability to choose among several scenarios allows for more freedom to build the ship hull, so that issues related to costs can be duly considered. Figure 10 reports the final hull shapes obtained using ORTHOinit (left) and ORTHOinit+ (right). Similarly, Figure 11 reports the wave pattern generated by the hull obtained using ORTHOinit (up) and ORTHOinit+ (down): ORTHOinit+ shows an attenuated diverging bow wave, which is indeed a confirmation of the slightly better value of Δ_f in Figure 9, obtained adopting ORTHOinit+.

Finally, combining the effects of ORTHOinit and ORTHOinit+ (i.e. the grey dotted line of ORTHOinit# in Figure 9a) we improved efficiency with respect to ORTHOinit+, and we still retained a dense solution, with again a remarkable difference in the 2nd and 3rd unknowns. This gives further evidence that the density issues studied in Section 6, which are a core subject of this paper, can play a noteworthy role. Also observe that HSS is on the overall less efficient than ORTHOinit#.

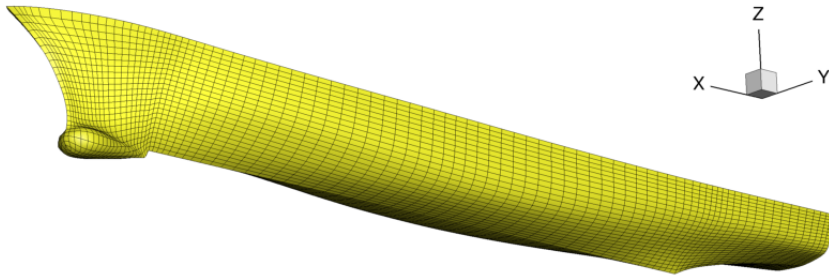


Fig. 8 Computational grid g_0 used for numerical implementation of the generalized KLE and for the potential flow solution.

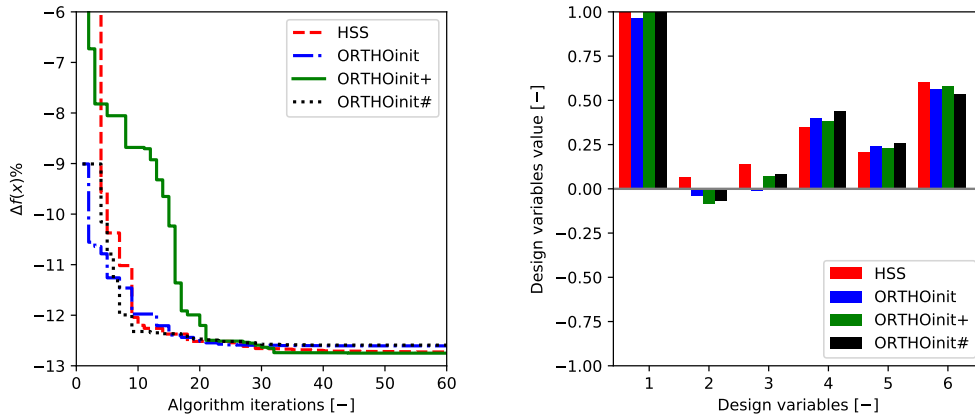


Fig. 9 Optimization results: DPSO convergence for the four initializations (left) and corresponding design vectors (right).

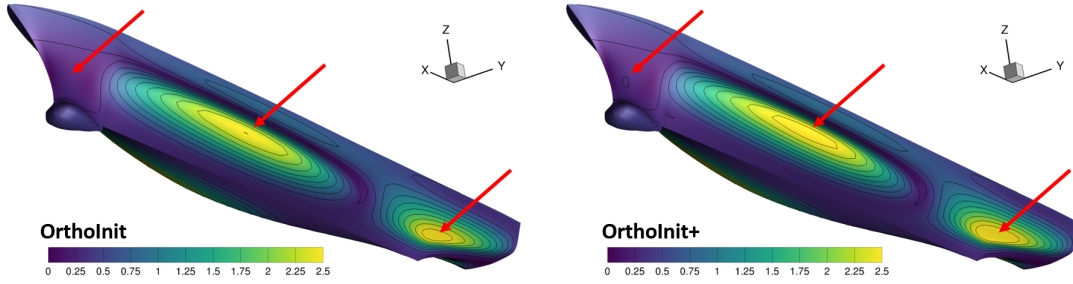


Fig. 10 Optimization results: the final hull obtained using ORTHOinit (left) and ORTHOinit+ (right). Red arrows indicate the differences.

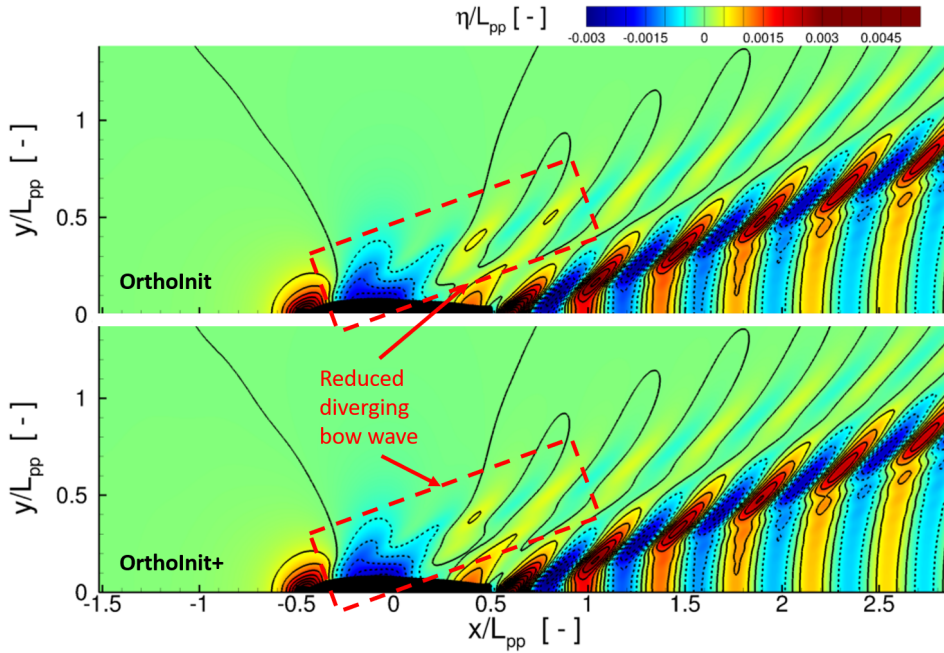


Fig. 11 Optimization results: the final wave pattern generated by the hull obtained using ORTHOinit (top) and ORTHOinit+ (bottom). Dashed boxes evidence the differences.

8 Conclusions and future work

In this paper we have analyzed novel initializations for DPSO, in order to better exploit the topology of the swarm in \mathbb{R}^n , and speed up in the early iterations the solution of unconstrained optimization and bound-constrained optimization problems.

Unlike the proposal by [6], the theory in the current paper yields a guideline for the choice of $2n$ particles' initial position/velocity, and not just n . Moreover, our initialization is dense and tends to scatter the particles in the search space. The latter fact was expected to provide a more powerful tool (as numerical results seem to confirm) for widely exploring the search space. Furthermore, our theoretical achievements yield a particles' initialization in DPSO which is related to (i.e. parametrized with) the real space of dimension n . Though no specific conclusion seems to be drawn by the latter observation, we remark that most of the exact derivative-free methods for smooth problems, as well as gradient-based methods for continuously differentiable functions, show noteworthy analogies. Indeed, for instance in the case of exact derivative-free methods, typically the use of search directions parallel to all the n coordinate axes helps improving efficiency, and is definitely indispensable to prove global convergence. On the other hand, observe that the gradient naturally summarizes the sensitivity of the function along the n coordinate axes, so that also gradient-based search directions rely on information referred to the n coordinate axes.

In case the parameter ω in (12) possibly adaptively changes at each iteration, the theory above does not hold anymore and it should be reformulated from scratch. The circumstance under which the parameter ω changes at each iteration is a key-aspect in several cases, when PSO limits arise and the progress of the algorithm is very slow. An example is provided when *stagnation* arises, i.e. when eventually the best particle of the swarm does not change its position. Unfortunately, the proposal in this paper is unable to cope with the last relevant issue, which may have a dramatic impact. Therefore, we are persuaded that such a limit of our proposal might possibly represent a future issue for a fruitful investigation. As a further limit of our theory, the reader may include our relative rigid choice of DPSO coefficients, which is imposed by Assumption 1. Indeed, we conjecture that a more general setting of DPSO parameters, where each particle retains its individual choice, may yield better results.

Finally, an adaptive criterion might be advisable, in order to restart the position and velocity of some particles after a given number of iterations. The adaptive criterion might for instance monitor the norm $\|\mathbf{X}_L(k)^{(j)}\|$, $j = 1, \dots, 2n$ (see also Section 5 of [6]), of the free response of particles. When the latter quantity approaches zero, a restart would re-impose orthogonality among the free responses of the particles, using the theory in Section 4.

Acknowledgements The work is supported by the US Department of the Navy Office of Naval Research Global, NICOP grant N62909-18-1-2033, administered by Dr. Woei-Min Lin, Dr. Elena McCarthy, and Dr. Salahuddin Ahmed, and by the Italian Flagship Project RITMARE, funded by the Italian Ministry of Education.

Appendix

Proof of Proposition 3

By (42), (43) and (46), along with a simple computation, we have

$$\det(\mathbf{w}_1 \dotscots \mathbf{w}_{2n}) = \det(\mathbf{z}_1 \dotscots \mathbf{z}_{2n})\eta,$$

where

$$\eta = \det \left[\begin{array}{ccccc|ccccc} 1 & -\alpha & \cdots & \cdots & -\alpha & -\delta & -\delta & \cdots & \cdots & -\delta \\ -\alpha & 1 & -\alpha & \cdots & \vdots & -\delta & \ddots & & & \vdots \\ \vdots & -\alpha & 1 & \ddots & \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\alpha & \vdots & & & \ddots & -\delta \\ -\alpha & \cdots & \cdots & -\alpha & 1 & -\delta & \cdots & \cdots & -\delta & -\delta \\ \hline & & & & & 1 & -2/(n-2) & \cdots & \cdots & -2/(n-2) \\ & & & & & -2/(n-2) & 1 & -2/(n-2) & \cdots & \vdots \\ & \mathbf{0} & & & & \vdots & -2/(n-2) & 1 & \ddots & \vdots \\ & & & & & \vdots & \vdots & \ddots & \ddots & -2/(n-2) \\ & & & & & -2/(n-2) & \cdots & \cdots & -2/(n-2) & 1 \end{array} \right] \quad (53)$$

and since the vectors $\mathbf{z}_1, \dots, \mathbf{z}_{2n}$ are orthonormal, by (53) we have

$$\det(\mathbf{w}_1 \dotscots \mathbf{w}_{2n}) = \det(\mathbf{M}_1) \det(\mathbf{M}_2), \quad (54)$$

being

$$\mathbf{M}_1 = \begin{bmatrix} 1 & -\alpha & \cdots & \cdots & -\alpha \\ -\alpha & 1 & -\alpha & \cdots & \vdots \\ \vdots & -\alpha & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\alpha \\ -\alpha & \cdots & \cdots & -\alpha & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad (55)$$

$$\mathbf{M}_2 = \begin{bmatrix} 1 & -2/(n-2) & \cdots & \cdots & -2/(n-2) \\ -2/(n-2) & 1 & -2/(n-2) & \cdots & \vdots \\ \vdots & -2/(n-2) & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -2/(n-2) \\ -2/(n-2) & \cdots & \cdots & -2/(n-2) & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (56)$$

Since \mathbf{M}_1 and \mathbf{M}_2 have the same pattern, the computation of $\det(\mathbf{M}_1)$ and $\det(\mathbf{M}_2)$ has the same difficulty. Thus, we focus without loss of generality on computing $\det(\mathbf{M}_1)$, whose value is invariant after subtracting the $(i+1)$ -th column to the i -th column of \mathbf{M}_1 , for $i = 1, \dots, n-1$. After the latter arrangement we obtain $\det(\mathbf{M}_1) = \det(\mathbf{\Gamma}_n)$, where (only the nonzero entries are reported)

$$\mathbf{\Gamma}_n = \begin{bmatrix} 1 + \alpha & & & & -\alpha \\ -(1 + \alpha) & 1 + \alpha & & & \vdots \\ & -(1 + \alpha) & \ddots & & \vdots \\ & & \ddots & 1 + \alpha & -\alpha \\ & & & -(1 + \alpha) & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (57)$$

Computing $\det(\mathbf{\Gamma}_n)$ by the first row and using induction, recalling that $\mathbf{\Gamma}_1 = 1$, we have

$$\det(\mathbf{\Gamma}_1) = 1$$

$$\det(\mathbf{\Gamma}_2) = (1 + \alpha) - \alpha(1 + \alpha) = 1 - \alpha^2$$

$$\det(\mathbf{\Gamma}_3) = (1 + \alpha)^2 + (1 + \alpha)[- \alpha(1 + \alpha) - \alpha(1 + \alpha)] = (1 + \alpha)^2 - 2\alpha(1 + \alpha)^2 = (1 + \alpha)^2(1 - 2\alpha)$$

\vdots

$$\det(\mathbf{\Gamma}_h) = (1 + \alpha)^{h-1}[1 - (h-1)\alpha],$$

so that by (57) we have for $\det(\mathbf{\Gamma}_{h+1})$, $h < n$

$$\det(\mathbf{\Gamma}_{h+1}) = \det \left[\begin{array}{c|ccc} 1 + \alpha & 0 & \cdots & 0 & -\alpha \\ \hline -(1 + \alpha) & & & & \\ 0 & & & & \\ \vdots & & & & \\ 0 & & & & \end{array} \mathbf{\Gamma}_h \right].$$

Solving again with respect to the first row, we obtain

$$\begin{aligned} \det(\mathbf{\Gamma}_{h+1}) &= (1 + \alpha) \det(\mathbf{\Gamma}_h) + (-1)^{h+2}(-\alpha)[- (1 + \alpha)]^h \\ &= (1 + \alpha)^h [1 - (h-1)\alpha] + (-1)^{2h+3}\alpha(1 + \alpha)^h \\ &= (1 + \alpha)^h [1 - (h-1)\alpha - \alpha] = (1 + \alpha)^h (1 - h\alpha). \end{aligned}$$

Thus, by (53), (54), (55) and (56)

$$\begin{aligned}\det(\mathbf{w}_1 \cdots \mathbf{w}_{2n}) &= \{(1 + \alpha)^{n-1} [1 - (n - 1)\alpha]\} \left\{ \left(1 + \frac{2}{n-2}\right)^{n-1} \left[1 + (n-1)\frac{2}{n-2}\right] \right\} \\ &= (1 + \alpha)^{n-1} [1 - (n - 1)\alpha] \frac{(3n - 4)n^{n-1}}{(n - 2)^n}.\end{aligned}$$

◇

References

1. C. AUDET, A. CUSTODIO, AND J. DENNIS JR., *Erratum: Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on Optimization, 18 (2008).
2. C. AUDET AND J. DENNIS JR., *Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on Optimization, 17 (2006).
3. P. BASSANINI, U. BULGARELLI, E. F. CAMPANA, AND F. LALLI, *The wave resistance problem in a boundary integral formulation*, Surveys on Mathematics for Industry, 4 (1994), pp. 151–194.
4. M. R. BONYADI AND Z. MICHALEWICZ, *A locally convergent rotationally invariant particle swarm optimization algorithm*, Swarm Intelligence, 8 (2014), pp. 159–198.
5. E. F. CAMPANA, M. DIEZ, G. FASANO, AND D. PERI, *Initial Particles Position for PSO, in Bound Constrained Optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 112–119.
6. E. F. CAMPANA, G. FASANO, AND A. PINTO, *Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization*, Journal of Global Optimization, 48 (2010), pp. 347–397.
7. E. F. CAMPANA, G. LIUZZI, S. LUCIDI, D. PERI, V. PICCIALLI, AND A. PINTO, *New global optimization methods for ship design problems*, Optimization and Engineering, 10 (2009), pp. 533–555.
8. E. F. CAMPANA, D. PERI, Y. TAHARA, AND F. STERN, *Shape optimization in ship hydrodynamics using computational fluid dynamics*, Computer Methods in Applied Mechanics and Engineering, 196 (2006), pp. 634–651.
9. X. CHEN, M. DIEZ, M. KANDASAMY, Z. ZHANG, E. F. CAMPANA, AND F. STERN, *High-fidelity global optimization of shape design by dimensionality reduction, metamodels and deterministic particle swarm*, Engineering Optimization, 47 (2015), pp. 473–494.
10. Z. CHEN AND Y. WANG, *Two-sided coherent risk measures and their application in realistic portfolio optimization*, Journal of Banking & Finance, 32 (2008), pp. 2667–2673.
11. C. W. CLEGHORN AND A. P. ENGELBRECHT, *Particle Swarm Convergence: Standardized Analysis and Topological Influence*, Springer International Publishing, Cham, 2014, pp. 134–145.
12. ———, *Particle swarm variants: standardized convergence analysis*, Swarm Intelligence, 9 (2015), pp. 177–203.
13. M. CLERC, *Stagnation analysis in particle swarm optimization or what happens when nothing happens*, tech. rep., 2006.
14. M. CLERC AND J. KENNEDY, *The particle swarm - explosion, stability, and convergence in a multidimensional complex space*, IEEE Transactions on Evolutionary Computation, 6 (2002), pp. 58–73.
15. A. CONN, K. SCHEINBERG, AND L. VICENTE, *Introduction to Derivative-Free Optimization*, Society for Industrial and Applied Mathematics, 2009.
16. M. CORAZZA, G. FASANO, AND R. GUSSO, *Particle swarm optimization with non-smooth penalty reformulation, for a complex portfolio selection problem*, Applied Mathematics and Computation, 224 (2013), pp. 611–624.
17. C. W. DAWSON, *A practical computer method for solving ship-wave problems*, in Proceedings of the 2nd International Conference on Numerical Ship Hydrodynamics, Berkeley, 1977, pp. 30–38.
18. M. DIEZ, D. PERI, G. FASANO, AND E. F. CAMPANA, *Hydroelastic optimization of a keel fin of a sailing boat: a multidisciplinary robust formulation for ship design*, Structural and Multidisciplinary Optimization, 46 (2012), pp. 613–625.
19. M. DIEZ, A. SERANI, E. F. CAMPANA, S. VOLPI, AND F. STERN, *Design space dimensionality reduction for single- and multi-disciplinary shape optimization*, in 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization (MA&O), AVIATION 2016, Washington D.C., USA, June 13-17, 2016.
20. M. DIEZ, A. SERANI, C. LEOTARDI, E. F. CAMPANA, D. PERI, U. IEMMA, G. FASANO, AND 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.
21. G. FASANO AND M. ROMA, *Iterative computation of negative curvature directions in large scale optimization*, Computational Optimization and Applications, 38 (2007), pp. 81–104.
22. ———, *Preconditioning Newton–Krylov methods in nonconvex large scale optimization*, Computational Optimization and Applications, 56 (2013), pp. 253–290.
23. A. GRIEWANK, *Automatic Differentiation*, Princeton University Press, 2014.
24. G. GRIGOROPOULOS, E. CAMPANA, M. DIEZ, A. SERANI, O. GOREN, K. SARIÖZ, D. DANIŞMAN, M. VISONNEAU, P. QUEUTEY, M. ABDEL-MAKSOU, ET AL., *Mission-based hull-form and propeller optimization of a transom stern destroyer for best performance in the sea environment*, in VII International Conference on Computational Methods in Marine Engineering MARINE2017, 2017.
25. M. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–435.
26. J. KENNEDY AND R. EBERHART, *Particle swarm optimization*, in Proceedings of the Fourth IEEE Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.
27. T. G. KOLDA, R. M. LEWIS, AND V. TORCZON, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Review, 45 (2003), pp. 385–482.
28. C. LEOTARDI, E. F. CAMPANA, AND M. DIEZ, *Resistance reduction of a military ship by variable-accuracy metamodel-based multidisciplinary robust design optimization*, in Proceedings of VI International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS 2015, Venice, Italy, 18-20 May, 2015.
29. J. LIU, C. WU, G. WU, AND X. WANG, *A novel differential search algorithm and applications for structure design*, Applied Mathematics and Computation, 268 (2015), pp. 246–269.
30. J. LIU, H. ZHU, Q. MA, L. ZHANG, AND H. XU, *An artificial bee colony algorithm with guide of global & local optima and asynchronous scaling factors for numerical optimization*, Applied Soft Computing, 37 (2015), pp. 608–618.
31. K. M. MALAN AND A. P. ENGELBRECHT, *Algorithm comparisons and the significance of population size*, IEEE Xplore Digital Library, 2008, pp. 914–920.

32. C. K. MONSON AND K. D. SEPPI, *The Kalman Swarm*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 140–150.
33. A. OLIVIERI, F. PISTANI, A. AVANZINI, F. STERN, AND R. PENNA, *Towing tank, sinkage and trim, boundary layer, wake, and free surface flow around a naval combatant INSEAN 2340 model*, tech. rep., DTIC, 2001.
34. E. OZCAN, S. CAD, T. S. NO, AND C. K. MOHAN, *Particle swarm optimization: Surfing the waves*, in Proceedings of the Congress on Evolutionary Computation, IEEE Press, 1999, pp. 6–9.
35. U. PAQUET AND A. P. ENGELBRECHT, *Particle swarms for linearly constrained optimisation*, *Fundam. Inf.*, 76 (2007), pp. 147–170.
36. R. POLI, *The sampling distribution of particle swarm optimisers and their stability*, Tech. Rep. CSM-465, University of Essex, 2007.
37. H. SCHLICHTING AND K. GERSTEN, *Boundary-Layer Theory*, Springer-Verlag, Berlin, 2000.
38. A. SERANI AND 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), Nantes, France, 15-17 May 2017, pp. 48–59.
39. A. SERANI, M. DIEZ, E. F. CAMPANA, G. FASANO, D. PERI, AND U. IEMMA, *Globally convergent hybridization of particle swarm optimization using line search-based derivative-free techniques*, Vol. 585, Springer Verlag, 2015, pp. 25–47.
40. A. SERANI, M. DIEZ, J. WACKERS, M. VISONNEAU, AND F. STERN, *Stochastic shape optimization via design-space augmented dimensionality reduction and RANS computations*, in AIAA Scitech 2019 Forum, San Diego, California, USA, January 7-11, 2019.
41. A. SERANI, G. FASANO, G. LIUZZI, S. LUCIDI, U. IEMMA, E. F. CAMPANA, F. STERN, AND M. DIEZ, *Ship hydrodynamic optimization by local hybridization of deterministic derivative-free global algorithms*, *Applied Ocean Research*, 59 (2016), pp. 115–128.
42. A. SERANI, C. LEOTARDI, U. IEMMA, E. F. CAMPANA, G. FASANO, AND M. DIEZ, *Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems*, *Applied Soft Computing*, 49 (2016), pp. 313–334.
43. F. STERN, J. LONGO, R. PENNA, A. OLIVIERI, T. RATCLIFFE, AND H. COLEMAN, *International collaboration on benchmark CFD validation data for surface combatant DTMB model 5415*, in Proceedings of the Twenty-Third Symposium on Naval Hydrodynamics, Val de Reuil, France, September 17-22, 2000.
44. T. TERLAKY, M. ANJOS, AND S. AHMED, *Advances and Trends in Optimization with Engineering Applications*, MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics, 2017.
45. I. C. TRELEA, *The particle swarm optimization algorithm: convergence analysis and parameter selection*, *Information Processing Letters*, 85 (2003), pp. 317–325.
46. F. VAN DEN BERGH, *An Analysis of Particle Swarm Optimizers*, PhD thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.
47. F. VAN DEN BERGH AND A. P. ENGELBRECHT, *A study of particle swarm optimization particle trajectories*, *Information Sciences*, 176 (2006), pp. 937–971.
48. A. VAZ AND L. VICENTE, *A particle swarm pattern search method for bound constrained global optimization*, *Journal of Global Optimization*, 39 (2007), pp. 197–219.
49. ———, *Pswarm: A hybrid solver for linearly constrained global derivative-free optimization*, *Optimization Methods and Software*, 24 (2009), pp. 669–685.
50. T. T. WONG, W. S. LUK, AND P. A. HENG, *Sampling with Hammersley and Halton points*, *Journal of Graphics Tools*, 2 (1997), pp. 9–24.