Dynamic system analysis and initial particles position in Particle Swarm Optimization

Emilio F. Campana*, Giovanni Fasano*† and Antonio Pinto*
*National Research Institute for Ships and Marine Systems - INSEAN
Rome, Italy 00128,
Email: (e.campana,g.fasano,a.pinto)@insean.it
†Dipartimento di Informatica e Sistemistica “A. Ruberti”,
Università di Roma “La Sapienza”,
Rome, Italy 00185
Email: fasano@dis.uniroma1.it

***Abstract*— This paper focuses on a solution technique for global optimization problems, where the objective function value is possibly computed by the numerical solution of a PDE system. The nature of these optimization problems is that of a ‘black-box’ type, where expensive simulations provide information to the optimizer, and each function evaluation could require several CPU-hours. The paper considers the evolutionary Particle Swarm Optimization (PSO) algorithm, for the minimization of a nonlinear function in the global optimization frameworks described. We reformulate the standard iteration of PSO [10], [3] into a linear dynamic system. Then, the latter is investigated in order to provide indications for the assessment of the initial particles position. We carry out our analysis on a generalized PSO iteration, which includes the standard one proposed in the literature. Therefore, our results perfectly apply to standard PSO too, without any modifications. In our scheme the path of any particle is possibly affected by the trajectories of all the other particles in the swarm. Our preliminary numerical experience, over a set of 35 standard test problems from the literature, confirms the theoretical analysis.**

I. Introduction

Many applications of optimization in engineering and science have to deal with extremely large computational requirements and often the cost of computing the objective function values or derivatives is so large that standard optimization methods cannot produce results in reasonable time or at reasonable cost. Engineering design is among the primary examples of this kind of application. Real-world shape optimization problems have to face complex geometry and typically involve a large number of variables, requiring hundreds or thousands of function evaluations to converge to an optimal design. If high-fidelity analysis[1] is required, both derivative-free and gradient-based optimization methods become more and more expensive.

Albeit the use of Global Optimization (GO) algorithms inevitably leads to a further increase of the computational effort, design engineers of marine, aeronautical, automotive transport systems, are very much tempted by taking this direction. Indeed, the margin for design improvements is continuously narrowing for the design engineers are producing near optimal configurations in many industrial fields, and the probability that performances breakthrough could come from local optimization methods is likely small. These issues are motivating our interest in the development of GO algorithms with reduced computational effort in terms of function evaluations.

This paper is concerned with the use of the PSO algorithm for the solution of the global optimization problem

$$\min_{x \in \mathcal{L}} f(x), \qquad \mathcal{L} \subset {\rm I\!R}^n, \tag{1}$$

where $f : {\rm I\!R}^n \to {\rm I\!R}$ is a *continuous function*, and $\mathcal{L}$ is a *closed* and *bounded* set. In particular, we aim at detecting a global minimum x^* of (1), such that $f(x^*) \leq f(x)$ for any $x \in \mathcal{L}$, and the function is computationally costly. A specific aspect of the paper is that we assume that, given the computational cost of the objective function, we can only afford a small number of particles in the swarm.

As well known, unlike local optimization frameworks, there are not optimality conditions which can characterize the point x^*. Indeed, no stopping criterion both theoretical and practical, may be adopted in order to detect the global minima of problem (1). Anyway, several approaches have been proposed in the literature for the solution of (1) (see [14] and therein references), either adopting deterministic [7] and/or stochastic [6] techniques.

The use of the latter techniques often requires the exploration over a possibly dense subset of $\mathcal{L}$; thus, the increasing computational burden would be definitely unaffordable for the shape design problems described above.

In the last decades both deterministic and stochastic methods for problem (1) have included many contributions, whose rationale is often suggested by biological systems and/or social behavioural interpretations.

In this paper we consider the PSO algorithm [10], an iterative method for global optimization, in the class of evolutionary algorithms. The latter scheme was originally conceived in ’95 as a model to describe the behaviour of a flock of

[1] E.g. numerical codes for the solution of the Navier-Stokes PDE, governing the flow around cars, airplanes or ships. In this case the CPU time for one function evaluation may be of the order of ten hours on current platforms, depending on the geometry details.

birds. Then, it has been recently improved (see also [3], [13], [17], [19], [2]) and is widely adopted within specific global optimization applications [1], [5], [15], [18].

The growing interest for PSO algorithm, to solve distinctive global optimization problems (e.g. ship design), is encouraged by the following appealing features:

- *balance* between the computation involved and the precision of the solution detected;
- constant *computational cost* and *memory engagement* at each iteration;
- availability of a *current approximate solution*;
- *derivatives* of the objective function not required;
- easy *implementation* and *parallelization* of the method.

We recall that PSO iteration is neither able to guarantee the convergence to a global minimum nor to a local minimum. Indeed, PSO is a heuristic method, and its reformulations in the literature are heuristics as well. The general PSO iteration can be described at step $k \geq 0$ by

$$x_j^{k+1} = x_j^k + d_j^k, \qquad j = 1, \ldots, P, \tag{2}$$

where $x_j^k \in \mathcal{L}$ is the current position of the j-th individual (*particle*) of the population (*swarm*), $d_j^k \in \mathbb{R}^n$ is the search direction ('d' usually stands for 'direction' in optimization frameworks) and x_j^{k+1} is the new position at step $k+1$. Observe that d_j^k may *not* be a descent direction for the objective function $f(x)$ at x_j^k. In particular, direction d_j^k depends on both the search direction d_j^{k-1} and the paths of the particles. The original version of PSO in [10] considers at step k the direction

$$d_j^k = d_j^{k-1} + \alpha^k (x_j^{\text{best}} - x_j^k) + \beta^k (x^{\text{best}} - x_j^k), \tag{3}$$

where α^k, β^k are suitable random scalars and

$$x_j^{\text{best}} = \underset{0 \leq h \leq k}{\operatorname{argmin}} \; \{f(x_j^h)\}, \qquad j = 1, \ldots, P$$

$$x^{\text{best}} = \underset{0 \leq h \leq k, \; j=1,\ldots,P}{\arg\min} \{f(x_j^h)\}.$$

In this paper we figure out a partial analysis of the particles trajectories, on a very general reformulation of PSO iteration. Our approach is not completely new in the literature (similar issues are partially considered and investigated in [12], [3], [10], [13]). In particular, we reformulate our PSO generalized iteration into a dynamic linear system, where feedback is not included. The lack of a feedback block is a simplification, motivated by a couple of basic reasons. On one hand, though preferable, a complete stability analysis of a closed-loop system would be far from our purposes. Furthermore, some practical results on the assessment of the initial particles position may be obtained, by only focusing on the free response of the open-loop system. We highlight that in our knowledge, the PSO literature has already provided indications about several practical aspects of PSO. Indeed, researchers and practitioners have been most concerned with issues like the choice of PSO's coefficients or how PSO compares with other evolutionary methods over optimization problems. Unfortunately, it seems that the qualitative and quantitative impact of a proper choice of the initial particles' position has not yet been deeply investigated.

A preliminary numerical experience on our generalized version of PSO algorithm confirms the hints of the theory.

In this paper $\mathbf{C}$ represents the set of complex numbers. With '$\mathrm{rk}(A)$' we indicate the rank of matrix A and '$\det(A)$' is its determinant. Finally we denote with I the identity matrix of suitable dimension.

In Section II we describe our generalized PSO iteration by means of a dynamic linear system, whose properties are partially analyzed in Sections III and III-A. Section IV is devoted to investigate promising starting points for the particles.

II. A GENERALIZED PSO MODEL

Consider the following (generalized) iteration of PSO algorithm in (2)-(3), which is reported by using a common notation adopted in the PSO literature ($k = 0, 1, \ldots$):

$$\begin{aligned} v_j^{k+1} &= \chi \left[w^k v_j^k + c_j r_j (p_j^k - x_j^k) + c_g r_g (p_g^k - x_j^k) \right], \\ x_j^{k+1} &= x_j^k + v_j^{k+1}, \end{aligned} \tag{4}$$

where $j = 1, \ldots, P$ indicates the j-th particle, P is finite, v_j^k and x_j^k are the *speed* and the *position* of particle j at step k, p_j^k and p_g^k respectively satisfy

$$\begin{aligned} p_j^k &= \operatorname{argmin}_{l \leq k} \left\{ f(x_j^l) \right\}, \\ p_g^k &= \operatorname{argmin}_{l \leq k, \; j=1,\ldots P} \left\{ f(x_j^l) \right\}, \end{aligned} \tag{5}$$

and $\chi, w^k, c_j, r_j, c_g, r_g$ are suitable bounded coefficients. Observe that the generalized relations (4) include both the cases in which only the *inertia* w^k or the *constriction* coefficient χ are used. *We assume that* r_j *and* r_g *are uniformly distributed random parameters* with $0 \leq r_j \leq 1$ and $0 \leq r_g \leq 1$.

Then, we can further generalize (4) by assuming that possibly the speed v_j^{k+1} depends on all the terms $p_h^k - x_j^k$, $h = 1, \ldots, P$, and not only on vectors $p_j^k - x_j^k$, $p_g^k - x_j^k$. In words this corresponds to allow a more general *social* contribution in the PSO iteration. The new iteration is therefore ($k = 0, 1, \ldots$)

$$\begin{aligned} v_j^{k+1} &= \chi_j^k \left[w_j^k v_j^k + \sum_{h=1}^{P} c_{h,j}^k r_{h,j}^k (p_h^k - x_j^k) \right], \\ x_j^{k+1} &= x_j^k + v_j^{k+1}, \end{aligned} \tag{6}$$

where $c_{h,j}^k$ and $r_{h,j}^k$ depend on the step (k), the current particle (j) and the other particles (h).

Without loss of generality at present we focus on the j-th particle and omit the subscript in the recurrence (6). Moreover, at present time we assume in (6) $\chi_j^k = \chi$, $c_{h,j}^k = c_h$, $r_{h,j}^k = r_h$ and $w_j^k = w$, for any $k \geq 0$. This is a common hypothesis in the PSO literature, which will be removed in Section IV. With the latter position the iteration (6) is equivalent to the *dynamic discrete, linear* and *stationary (time-invariant) system*

$$X(k+1) = \begin{pmatrix} \chi w I & -\sum_{h=1}^{P} \chi c_h r_h I \\ \chi w I & \left(1 - \sum_{h=1}^{P} \chi c_h r_h\right) I \end{pmatrix} X(k) \ + \begin{pmatrix} \sum_{h=1}^{P} \chi c_h r_h p_h^k \\ \sum_{h=1}^{P} \chi c_h r_h p_h^k \end{pmatrix}, \qquad (7)$$

where

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

The authors are aware that (7) may be inappropriate for a full analysis of stability, since the vector p_h^k is substantially a feedback term of the system (indeed p_h^k depends on the sequences $\{x_h^k\}$). However, as described in the sequel, modelling the feedback of system (7) (i.e. modelling the expression of p_h^k) is more or less equivalent to resort to a deep knowledge of the objective function $f(x)$. Unfortunately, the latter is only known at points x_h^k. Nonetheless, instead of studying the ambitious issue of the stability, for a closed-loop model of system (7), we aim at considering the more tractable free response of (7) (see also [3]). Then, the latter is used to obtain some partial indications on the assessment of the starting positions of the particles in PSO.

The sequence $\{X(k)\}$ identifies a trajectory in the state space $\mathbb{R}^{2n}$, and since (7) is a linear and stationary system, we may consider the *free response* $X_{\mathcal{L}}(k)$ and the *forced response* $X_{\mathcal{F}}(k)$ of the trajectory $\{X(k)\}$. Then, considering (7) we explicitly obtain at step $k \geq 0$ [16]

$$X(k) = X_{\mathcal{L}}(k) + X_{\mathcal{F}}(k), \qquad (8)$$

where

$$X_{\mathcal{L}}(k) = \Phi(k) X(0), \qquad X_{\mathcal{F}}(k) \ = \sum_{\tau=0}^{k-1} H(k-\tau) U(\tau), \ (9)$$

and (after few calculation)

$$\Phi(k) \ = \ \begin{pmatrix} \chi w I & -\sum_{h=1}^{P} \chi c_h r_h I \\ \chi w I & \left(1 - \sum_{h=1}^{P} \chi c_h r_h\right) I \end{pmatrix}^{k}, \ (10)$$

$$H(k-\tau) \ = \ \begin{pmatrix} \chi w I & -\sum_{h=1}^{P} \chi c_h r_h I \\ \chi w I & \left(1 - \sum_{h=1}^{P} \chi c_h r_h\right) I \end{pmatrix}^{k-\tau-1} \ (11)$$

$$U(\tau) \ = \ \begin{pmatrix} \sum_{h=1}^{P} \chi c_h r_h p_h^{\tau} \\ \sum_{h=1}^{P} \chi c_h r_h p_h^{\tau} \end{pmatrix}. \qquad (12)$$

Observe that $X_{\mathcal{L}}(k)$ in (9) does not depend on the vector p_h^k, but uniquely on the initial point $X(0)$. On the contrary, $X_{\mathcal{F}}(k)$ in (9) depends on the vector p_h^k and is independent of $X(0)$. The latter practical observation allows us to compute separately the two terms. In order to carry out our conclusions, in the next two sections we compute $X_{\mathcal{L}}(k)$ by investigating the eigenpairs of matrix $\Phi(k)$ in (10).

III. Computation of the free response $X_{\mathcal{L}}(k)$

As we addressed in Section II, the computation of the trajectory $\{X(k)\}$ in (8) strongly depends on the sequences $\{p_h^k\}$, $h = 1, \ldots, P$. Furthermore we know by definition that asymptotically

$$\lim_{k \to \infty} X(k) = \lim_{k \to \infty} X_{\mathcal{F}}(k),$$

i.e. the free response $X_{\mathcal{L}}(k)$ is effective only for finite values of k. Nevertheless, in the Introduction we reported some classes of shape design problems, where the latter feature is relevant, since the computational resources only allow modest values of k. In this section we focus on the properties of $X_{\mathcal{L}}(k)$, so that it can be used to properly define the starting point of each particle.

Let us consider the following position in (10) and (11)

$$a = \chi w, \qquad \omega = \sum_{h=1}^{P} \chi c_h r_h, \qquad (13)$$

so that for any $\lambda \in \mathbf{C}$

$$\Phi(1) - \lambda I = \begin{pmatrix} (a-\lambda) I & -\omega I \\ a I & (1 - \omega - \lambda) I \end{pmatrix},$$

whose Schur complement is given by $(1 - \omega - \lambda + a\omega/(a-\lambda))I$. We aim at computing the $2n$ eigenvalues $\lambda^{(1)}, \ldots, \lambda^{(2n)}$ of $\Phi(1)$; thus, assuming for a while $\lambda^{(l)} \neq a$, $l = 1, \ldots, 2n$, we obtain after few calculation involving the Schur complement

$$\det\left(\Phi(1) - \lambda I\right) \ = \ \left[\lambda^2 - (1 - \omega + a)\lambda + a\right]^n. \ (14)$$

Therefore, $\Phi(1)$ has at most two distinct eigenvalues, each with algebraic multiplicity n, i.e. $\lambda^{(1)} = \cdots = \lambda^{(n)} = \lambda_1$ and $\lambda^{(n+1)} = \cdots = \lambda^{(2n)} = \lambda_2$ with

$$\lambda_1 = \frac{1 - \omega + a - \left[(1 - \omega + a)^2 - 4a\right]^{1/2}}{2}$$
$$\lambda_2 = \frac{1 - \omega + a + \left[(1 - \omega + a)^2 - 4a\right]^{1/2}}{2}. \qquad (15)$$

Observe that either $\lambda_1 = a$ or $\lambda_2 = a$ *if and only if* $a = 0$ or $\omega = 0$. However, $a \neq 0$ and $\omega \neq 0$ is a very common assumption in PSO literature [3], [10]. Thus, without loss of generality $\lambda_1 \neq a$ and $\lambda_2 \neq a$, and in accordance with the literature we may consider the following.

Assumption 3.1: We assume in (13) $a \neq 0$ and $\omega > 0$.

Moreover, $p_\Phi(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)$ is the minimal polynomial of matrix $\Phi(1)$ and $\det[\Phi(1)] = a^n$, so that for $n = 1$ we obtain $\det[\Phi(1)] = \lambda_1 \lambda_2 = a$. By considering the quantity $\Delta = (1-\omega+a)^2 - 4a$ in (15), the following cases are analyzed:

- $a < 0$ which yields $\Delta > 0$, thus λ_1, λ_2 are *real* and *distinct* (one is positive and the other is negative);
- $a > 0$ which yields condition $\Delta \geq 0$ as long as (see Assumption 3.1)

$$0 < \omega \leq (1-\sqrt{a})^2 \qquad \text{or} \qquad \omega \geq (1+\sqrt{a})^2, \quad (16)$$

  which generates the following subcases:

  (1) $0 < \omega < (1-\sqrt{a})^2 \implies \lambda_1$ and λ_2 are real, distinct and both positive;

  (2) $\omega = (1-\sqrt{a})^2$ or $\omega = (1+\sqrt{a})^2 \implies$ $\lambda_1 = \lambda_2 = \dfrac{1-\omega+a}{2} = \pm\sqrt{a}$;

  (3) $\omega > (1+\sqrt{a})^2 \implies \lambda_1$ and λ_2 are real, distinct and both negative.

To summarize the previous considerations in terms of dynamic analysis [16] we have:

- if $a > 0$ and $(1-\sqrt{a})^2 < \omega < (1+\sqrt{a})^2$ then λ_1 and λ_2 are complex conjugate, so that they generate natural pseudoperiodic modes (this generalizes the results in [9]). If $a > 0$ and $0 < \omega < (1-\sqrt{a})^2$ then we have exactly $2n$ natural aperiodic modes. Finally if $a > 0$ and $\omega > (1+\sqrt{a})^2$ then we have exactly $2n$ natural alternating modes;
- if $a < 0$ we have exactly n natural aperiodic modes and n natural alternating modes.

According with (9) [16],

$$X_{\mathcal{L}}(k) = \Phi(k)X(0) = \Phi(1)^k X(0)$$

$$= \begin{pmatrix} aI & -\omega I \\ aI & (1-\omega)I \end{pmatrix}^k X(0), \qquad (17)$$

and from Assumption 3.1 $\lambda_q \neq a$, $q = 1, 2$, then

$$\mathrm{rk}\,[\Phi(1) - \lambda_q I] =$$

$$\mathrm{rk}\left[\begin{pmatrix} I & 0 \\ \dfrac{a}{a-\lambda_q}I & I \end{pmatrix}\begin{pmatrix} (a-\lambda_q)I & -\omega I \\ 0 & 0 \end{pmatrix}\right] = n,$$

which implies that the *algebraic multiplicity* and the *geometric multiplicity* of eigenvalues λ_1 and λ_2 coincide with n, i.e. the unsymmetric matrix $\Phi(1)$ is *diagonalizable* and a basis of $2n$ real eigenvectors of $\Phi(1)$ exists.

A. Computing the eigenvectors of $\Phi(1)$

According with the last considerations of Section III, $2n$ eigenvectors of matrix $\Phi(1)$ exist and without loss of generality they have the form

$$v_i = \begin{pmatrix} v_i^1 \\ v_i^2 \end{pmatrix}, \qquad v_i^1, v_i^2 \in \mathbb{R}^n, \quad i = 1, \ldots, 2n.$$

Then, to compute the eigenvectors of $\Phi(1)$ we impose relations

$$0 = \left[\Phi(1) - \lambda^{(i)} I\right] v_i =$$

$$\begin{pmatrix} I & 0 \\ \frac{a}{a-\lambda^{(i)}}I & I \end{pmatrix}\begin{pmatrix} (a-\lambda^{(i)})I & -\omega I \\ 0 & 0 \end{pmatrix}\begin{pmatrix} v_i^1 \\ v_i^2 \end{pmatrix},$$

so that from Assumption 3.1 we explicitly obtain the matrix V of the $2n$ eigenvectors $V = [v_1 \cdots v_n v_{n+1} \cdots v_{2n}] \in \mathbb{R}^{2n \times 2n}$, with

$$V = \begin{pmatrix} I & I \\ \dfrac{a-\lambda_1}{\omega}I & \dfrac{a-\lambda_2}{\omega}I \end{pmatrix}.$$

These eigenvectors are linearly independent, provided that $\lambda_1 \neq \lambda_2$; the latter condition is not restrictive for our purposes, therefore from (16) we consider the following.

Assumption 3.2: We assume in (13) $\omega \neq (1-\sqrt{a})^2$ and $\omega \neq (1+\sqrt{a})^2$ for any $a > 0$, so that the $2n$ eigenvectors of matrix $\Phi(1)$ are linearly independent.

Therefore we have $V^{-1}\Phi(1)V = \Lambda$, with

$$\Lambda = \begin{pmatrix} \lambda_1 I & \\ & \lambda_2 I \end{pmatrix} \in \mathbb{R}^{2n \times 2n}$$

and after few calculation

$$V^{-1} = \frac{1}{\lambda_1 - \lambda_2}\begin{pmatrix} (a-\lambda_2)I & -\omega I \\ -(a-\lambda_1)I & \omega I \end{pmatrix}.$$

The previous results yield from (17)

$$\begin{aligned} X_{\mathcal{L}}(k) &= V\Lambda^k V^{-1} X(0) \\ &= \frac{1}{\lambda_1 - \lambda_2}\sum_{i=1}^{n}\left[\lambda_1^k X(0)^T\left((a-\lambda_2)e_i - \omega e_{n+i}\right)v_i - \right. \\ &\quad \left. \lambda_2^k X(0)^T\left((a-\lambda_1)e_i - \omega e_{n+i}\right)v_{n+i}\right] \\ &= \frac{1}{\lambda_1 - \lambda_2}\sum_{i=1}^{n}\left[\lambda_1^k\left((a-\lambda_2)X(0)_i - \omega X(0)_{n+i}\right)v_i - \right. \\ &\quad \left. \lambda_2^k\left((a-\lambda_1)X(0)_i - \omega X(0)_{n+i}\right)v_{n+i}\right], \end{aligned}$$

where $e_i \in \mathbb{R}^{2n}$ is the unit vector with 1 in position i, and $X(0)_i$ is the i-th entry of $X(0)$, i.e.

$$X(0) = \begin{pmatrix} X(0)_1 \\ \vdots \\ X(0)_{2n} \end{pmatrix}.$$

TABLE I
THE COEFFICIENTS $\gamma_i(k)$, $i = 1, \ldots, 4$ IN (18).

$$\gamma_1(k) = \begin{cases} \dfrac{\lambda_1^k(a-\lambda_2) - \lambda_2^k(a-\lambda_1)}{\lambda_1 - \lambda_2} & \lambda_1, \lambda_2 \ \text{real} \\ \rho^{k+1}\dfrac{\sin k\theta}{\sin\theta} - \rho^k \dfrac{\sin(k-1)\theta}{\sin\theta} & \lambda_1, \lambda_2 \ \text{complex,} \end{cases}$$

$$\gamma_2(k) = \begin{cases} \dfrac{\omega(\lambda_1^k - \lambda_2^k)}{\lambda_1 - \lambda_2} & \lambda_1, \lambda_2 \ \text{real} \\ \omega\rho^{k-1}\dfrac{\sin k\theta}{\sin\theta} & \lambda_1, \lambda_2 \ \text{complex,} \end{cases}$$

$$\gamma_3(k) = \begin{cases} \dfrac{(\lambda_1^k - \lambda_2^k)}{\lambda_1 - \lambda_2}\dfrac{(a-\lambda_1)(a-\lambda_2)}{\omega} & \lambda_1, \lambda_2 \ \text{real} \\ \rho^k\dfrac{\sin k\theta}{\sin\theta}\left(\dfrac{\rho^2 - 2\rho\cos\theta + 1}{\omega}\right) & \lambda_1, \lambda_2 \ \text{complex,} \end{cases}$$

$$\gamma_4(k) = \begin{cases} \dfrac{\lambda_1^k(a-\lambda_1) - \lambda_2^k(a-\lambda_2)}{\lambda_1 - \lambda_2} & \lambda_1, \lambda_2 \ \text{real} \\ \rho^{k+1}\dfrac{\sin k\theta}{\sin\theta} - \rho^k \dfrac{\sin(k+1)\theta}{\sin\theta} & \lambda_1, \lambda_2 \ \text{complex.} \end{cases}$$

Furthermore, for $i = 1, \ldots, n$

$$\begin{cases} v_i = e_i + \dfrac{a-\lambda_1}{\omega} e_{n+i} \\ v_{n+i} = e_i + \dfrac{a-\lambda_2}{\omega} e_{n+i}, \end{cases}$$

then

$$\begin{aligned} X_{\mathcal{L}}(k) = \sum_{i=1}^{n} & \left[\gamma_1(k)X(0)_i e_i - \gamma_2(k)X(0)_{n+i}e_i + \right. \\ & \left. \gamma_3(k)X(0)_i e_{n+i} - \gamma_4(k)X(0)_{n+i}e_{n+i}\right], \quad (18) \end{aligned}$$

where coefficients $\gamma_i(k)$, $i = 1, \ldots, 4$ are given in Table I. Observe that the first value of each coefficient $\gamma_1(k), \ldots, \gamma_4(k)$ refers to real eigenvalues λ_1 and λ_2, while the second value holds in case λ_1 and λ_2 are conjugate (i.e. $\lambda_1 = \rho e^{-j\theta}$, $\lambda_2 = \rho e^{j\theta}$, with $a = \rho^2$). In the end, from (18) we obtain

$$\begin{aligned} X_{\mathcal{L}}(k) = \sum_{i=1}^{n} & \left[\gamma_1(k)X(0)_i - \gamma_2(k)X(0)_{n+i}\right] e_i + \\ & \left[\gamma_3(k)X(0)_i - \gamma_4(k)X(0)_{n+i}\right] e_{n+i}. \quad (19) \end{aligned}$$

Remark 3.1: Relation (19) suggests the following remarkable conclusions:

- the free response $X_{\mathcal{L}}(k)$ in (8) has the same formal expression for each particle, and is uniquely dependent on the initial point $X(0)$ of the particle;
- by simply imposing $X(0)_i = X(0)_{n+i} = 0$ in (19), the free response of a particle has zero entries on the i-th and $(n+i)$-th axis. Thus, according with the free response in (19), each particle's trajectory has nonzero projection on any subspace of $\mathbb{R}^{2n}$, provided that the initial point $X(0)$ is suitably chosen.

From (19) a suitable choice of the starting point $X(0)$ of any particle may guarantee an improved exploration of the state space. The latter issue is noteworthy and plays a key role within global optimization frameworks, in order to ensure the convergence to any global minimum [14].

IV. HINTS FOR THE STARTING POINT OF EACH PARTICLE

According with Remark 3.1, in this section we give some theoretical indications about the choice of the initial point $X(0)$ of each particle, which is a crucial issue in PSO. In particular, let the set $\mathcal{L}$ in (1) satisfy $\mathcal{L} = \{x \in \mathbb{R}^n : \|x\|_\alpha \leq L, \ \ L > 0, \ \ \alpha > 0\}$ and let us indicate $X(k)^{(j)}$ the trajectory of the j-th particle. Observe that the latter assumption on $\mathcal{L}$ is very common in real applications. From (8) $X(k)^{(j)}$ linearly depends on the contribution of the free response $X_{\mathcal{L}}(k)^{(j)}$, i.e. it depends on the initial point $X(0)^{(j)}$.

In this section we assume that exactly n particles compose the swarm, though with a similar reasoning our results may be extended to smaller/larger sets of particles. We aim at assessing the initial points $X(0)^{(j)}$, $j = 1, \ldots, n$, of the particles, in such a way that the state space $\mathbb{R}^{2n}$ is explored as widely as possible by the trajectories $\{X(k)^{(j)}\}$. This is a general and intriguing issue for global optimization algorithms, which often resort to randomly generated sequences. Here we propose a set of starting points $\{X(0)^{(j)}\}$ such that for any fixed index k, the sequence $\{X_{\mathcal{L}}(k)^{(j)}\}$ is *scattered* in the state space $\mathbb{R}^{2n}$. Our numerical experience confirms that the latter result may be often fruitfully used.
To the latter purpose we consider in (13) for any particle $j = 1, \ldots, n$ the parameters a_j, w_j in place of a, w. Accordingly, coefficients $\gamma_1(k), \ldots, \gamma_4(k)$ in (19) are given for the j-th particle by $\gamma_1(k)^{(j)}, \ldots, \gamma_4(k)^{(j)}$.

Consider the scalars $\alpha^{(j)}, \beta^{(j)} \in \mathbb{R}$, with $|\alpha^{(j)}| + |\beta^{(j)}| \neq 0$, and the vector $t_j \in \mathbb{R}^n$, $j = 1, \ldots, n$, such that

$$t_j = \frac{\sqrt{n}}{n}\sum_{i=1}^{n} e_i - \frac{\sqrt{n}}{2} e_j. \quad (20)$$

Then, let the vector $X(0)^{(j)}$ be given by

$$X(0)^{(j)} = \begin{pmatrix} \alpha^{(j)} t_j \\ \\ \beta^{(j)} t_j \end{pmatrix}, \qquad j = 1, \ldots, n. \quad (21)$$

Proposition 4.1: Consider the set of initial points $\{X(0)^{(j)}\}$ defined in (21). Consider in (13) for any particle j the parameters a_j, w_j in place of parameters a, w, and let the Assumption 3.2 hold. Then for any $k \geq 0$ the following relations hold

$$\left[X_{\mathcal{L}}(k)^{(j)}\right]^T X_{\mathcal{L}}(k)^{(h)} = 0, \qquad \text{for any} \ \ 1 \leq j \neq h \leq n. \quad (22)$$

Proof
From Assumption 3.2 relation (19) is defined. Thus, by simple

substitution from (20) and (21), we obtain

$$X_{\mathcal{L}}(k)^{(j)} = \begin{pmatrix} \left[\alpha^{(j)}\gamma_1(k)^{(j)} - \beta^{(j)}\gamma_2(k)^{(j)}\right] t_j \\ \\ \left[\alpha^{(j)}\gamma_3(k)^{(j)} - \beta^{(j)}\gamma_4(k)^{(j)}\right] t_j \end{pmatrix}$$

$$X_{\mathcal{L}}(k)^{(h)} = \begin{pmatrix} \left[\alpha^{(h)}\gamma_1(k)^{(h)} - \beta^{(h)}\gamma_2(k)^{(h)}\right] t_h \\ \\ \left[\alpha^{(h)}\gamma_3(k)^{(h)} - \beta^{(h)}\gamma_4(k)^{(h)}\right] t_h \end{pmatrix},$$

and since $t_j^T t_h = 0$, for $1 \leq j \neq h \leq n$, relations (22) hold.

We remark that the choice (20)-(21) generates at step k vectors $X_{\mathcal{L}}(k)^{(1)}, \ldots, X_{\mathcal{L}}(k)^{(n)}$, which form an orthogonal basis in the state subspace of the positions. In Figs 1-8 we show examples with $n = 2$ (i.e. $X_{\mathcal{L}}(k)^{(j)} \in \mathrm{I\!R}^4$, $j = 1, 2$, $k \geq 0$) and two particles. Particles # 1 and # 2 are sketched in the two dimensional subspace of positions with empty triangles and filled circles respectively (we set $a_1 = a_2 = 0.9$ and $\omega_1 = \omega_2 = 0.4$ in (13) for both particles). Examples are given for four different well-known test problems [4]: the 'Six-humps camel-back', the 'Cosine mixture', the 'Treccani' and the 'Shubert penalized 2'. On the left sides of the figures only the free responses $X_{\mathcal{L}}(k)^{(1)}$ and $X_{\mathcal{L}}(k)^{(2)}$ are drawn, while on the right sides the full trajectories $X(k)^{(1)}$ and $X(k)^{(2)}$ are reported. Moreover, in the top figures (Figs. 1,3,5,7) the initial points $X(0)^{(1)}$ and $X(0)^{(2)}$ are randomly chosen, while in bottom figures (Fig. 2,4,6,8), $X(0)^{(1)}$ and $X(0)^{(2)}$ are chosen as suggested in (21) (this explains the perfect orthogonality of the free responses). Observe that with the random choice the algorithm fails to converge, because the trajectories substantially move along the segment joining $X(0)^{(1)}$ and $X(0)^{(2)}$. On the contrary, in all the other cases (Fig. 2,4,6 and 8) the orthogonality of the free responses of the two particles (left side) determines the convergence (Particle #2 is attracted by Particle #1 but from (22) they do not overlap).

The latter result may be substantially interpreted as follows. At least for small values of parameter k, the choice (21) tends to preserve orthogonality among the trajectories. This helps the particles to be 'distributed' in the state space, inasmuch as for any k the set of positions $\{x_1^k, \ldots, x_P^k\}$ (see (6)) is likely an independent set.

In order to provide further numerical evidence of this finding, we selected a set of 35 well-known test problems (see [11]), with dimensions ranging from $n = 2$ through $n = 30$. Table II reports the problem name, dimension, and the value found by following the two different initialization strategies of the global minima points: $f(x^*)_{independent}$ according to 21) and $f(x^*)_{random}$ for the random choice, respectively. With the new strategy, the minimum found is improved in 31 out of 35 cases. Interestingly, the improvement does not seem to depend on the number of variables n. It is worthwhile to notice that in all the cases we set the number of Particles in the swarm equal to n. This is motivated by the fact that we are interested in swarm of small dimension, assuming that the objective function is expensive. Finally, we can readily prove that replacing the sequence $\{t_j\}$ in (20) with any orthogonal sequence of n nonzero vectors, the result of Proposition 4.1 still holds.

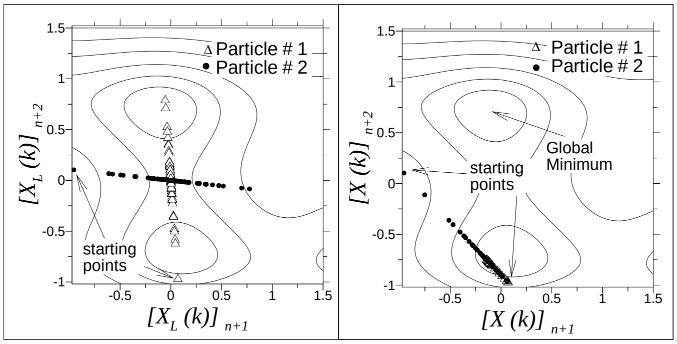

Fig. 1. Random choice of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (*'Six-humps camel-back'* function [4]). Particle # 1 and Particle # 2 substantially overlap: the global minimum is not detected.

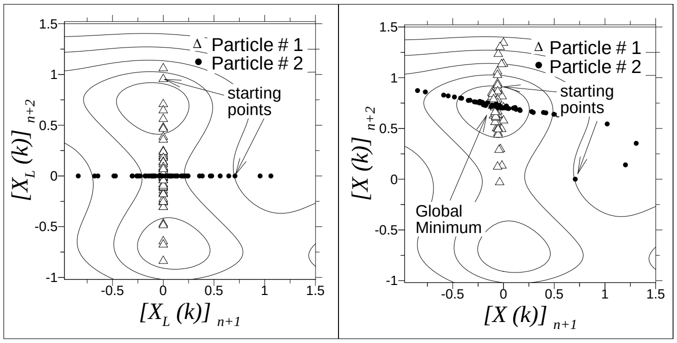

Fig. 2. Choice (21) of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (*'Six-humps camel-back'* function [4]). The two particles generate independent trajectories: the global minimum is detected.

V. Conclusions

With the aim of using PSO in GO problems where expensive analysis is involved, we explore the case in which a small number of particles are used (i.e. the number of particles is equal to the number of variables). A generalized version of Particle Swarm Optimization (PSO) algorithm is considered. In particular, using an open-loop model for a dynamic linear reformulation of PSO iteration, we assess the starting points of the swarm particles. A complete numerical experience is still necessary in order to evaluate the convenience of formula (20), with respect to other formulae aiming at yielding $t_j^T t_h = 0$, where $1 \leq j \neq h \leq n$. However, preliminary results on a set of 35 algebraic test function, with a number of variables from $n = 2$ to $n = 30$ look promising. Further numerical experience is still necessary in order to evaluate the optimum number of particles which must be used in the optimization process.

Acknowledgment

This work has been partially supported by the *Ministero delle Infrastrutture e dei Trasporti* in the framework of the research plan "Programma di Ricerca sulla Sicurezza", Decreto

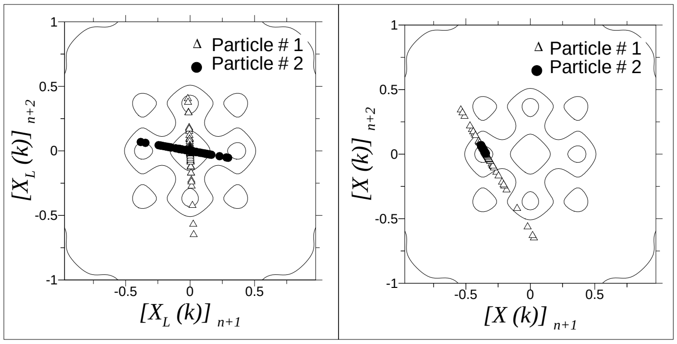

Fig. 3. Random choice of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (*‘Cosine mixture’* function [4]). Particle # 1 and Particle # 2 substantially overlap: the global minimum is not detected.

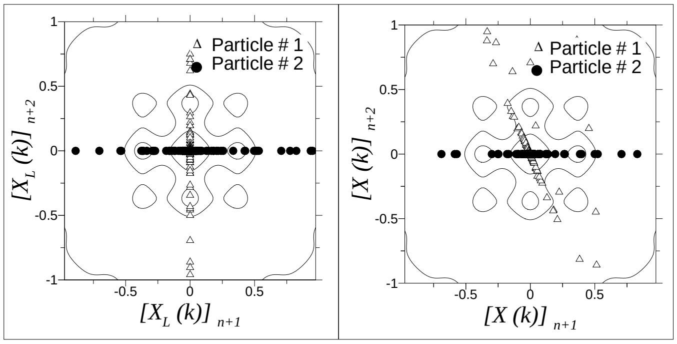

Fig. 4. Choice (21) of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (‘*Cosine mixture*’ function [4]). The two particles generate independent trajectories: the global minimum is detected.

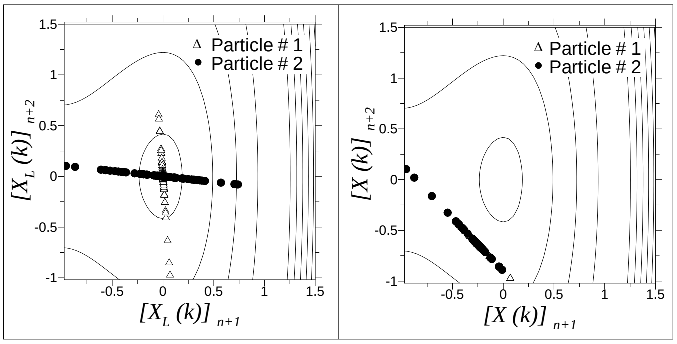

Fig. 5. Random choice of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (*‘Treccani’* function [4]). Particle # 1 and Particle # 2 substantially overlap: the global minimum is not detected.

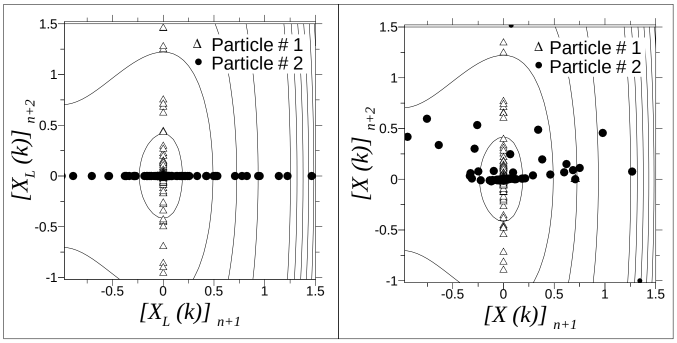

Fig. 6. Choice (21) of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (‘*Treccani*’ function [4]). The two particles generate independent trajectories: the global minimum is detected.

17/04/2003 G.U. n. 123 del 29/05/2003, and by the U.S. Office of Naval Research under the NICOP grant No. 000140210489 through Dr. Pat Purtell.

REFERENCES

[1] C.A. Baker, L.T. Watson, B. Grossman, R.T. Haftka, W.H. Mason, Parallel global aircraft configuration design space exploration, Proceedings of the 8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Paper No. 2000–4763–CP, Long Beach, California (USA), 2000.

[2] E.F. Campana, G. Liuzzi, S. Lucidi, D. Peri, V. Piccialli, A. Pinto, New Global Optimization Methods for Ship Design Problems, Technical Report INSEAN 2005-01.

[3] M. Clerc, J. Kennedy, The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1, 2002.

[4] L.C.W. Dixon, G.P. Szego, The optimization problem: An introduction, L.C.W.Dixon and G.P.Szego (Eds.), Towards Global Optimization II, New York: North Holland, 1978.

[5] P.C. Fourie, A.A. Groenwold, Particle Swarms in Size and Shape Optimization, Proceedings of the International Workshop on Multidisciplinary Design Optimization, Pretoria, South Africa, August 7-10, 2000, pp. 97-106.

[6] R. Horst, P.M. Pardalos, N.V. Thoai, Introduction to Global Optimization, Kluwer Academic publishers, The Netherlands, Dodrecht, 2000.

[7] R. Horst, H. Tuy, Global Optimization: deterministic approaches, Springer Verlag, Berlin, 1990.

[8] A. Jameson, J.J. Alonso, Future Research Avenues in Computational Engineering and Design, 4th International Congress on Industrial and Applied Mathematics (ICIAM 1999), Edinburgh, Scotland, July 5-9, 1999.

[9] J. Kennedy, Methods of agreement: Inference among the eleMentals, Proceedings of the 1998 IEEE International Symposium on Intelligent Control, pp. 883-887, 1998.

[10] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proceedings of the 1995 IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948, 1995.

[11] S. Lucidi, M. Piccioni, Random tunneling by means of acceptance-rejection sampling for global optimization, J. Optim. Theory and Appl., 62(2):255–279, 1989.

[12] C.K. Monson, K.D. Seppi, The Kalman Swarm - A New Approach to Particle Motion in Swarm Optimization, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), Seattle, Washington, pp. 140-150, 2004.

[13] E. Ozcan, C.K. Mohan Particle Swarm Optimization: Surfing the Waves, Proceedings of the 1999 IEEE Congress on Evolutionary Comnputation, IEEE Service Center, Piscataway, NJ, 1939-1944, 1999.

[14] J.D. Pinter, Global Optimization in Action. Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications, Kluwer Academic Publishers, The Netherlands, 1996.

[15] A. Pinto, D. Peri, E.F. Campana, Global Optimization Algorithms in Naval Hydrodynamics, Ship Technology Research, 51, 3, pp. 123-133, 2004.

[16] P.E. Sarachik, Principles of linear systems, Cambridge University Press, 1997.

[17] Y. Shi, R. Eberhart, Parameter Selection in Particle Swarm Optimization, The seventh Annual Conference on Evolutionary Computation, 1945-1950, 1998.

[18] G. Venter, J. Sobieszczanski-Sobieski, Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization Structural and Multidisciplinary Optimization, vol. 26, no. 1-2, pp. 121-131, 2004.

[19] Y.L. Zheng, L.H. Ma, L.Y. Zhang, J.X. Qian, *O*n the convergence analysis and parameter selection in particle swarm optimization, Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003.

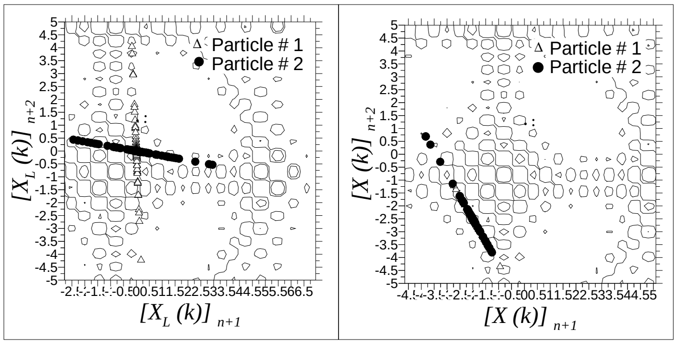

Fig. 7. Random choice of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (‘*Shubert penalized 2’* function [4]). Particle # 1 and Particle # 2 substantially overlap: the global minimum is not detected.

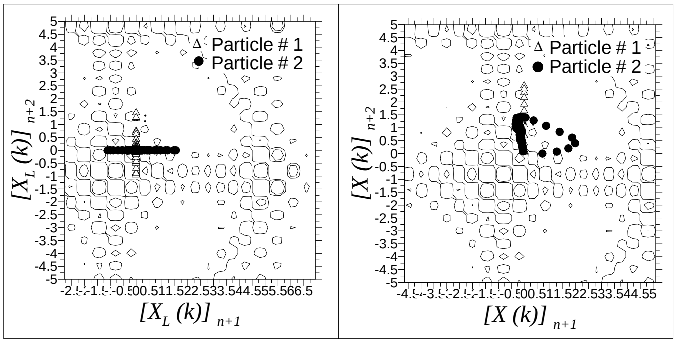

Fig. 8. Choice (21) of initial points $X(0)^j$, $j = 1, 2$, with $n = 2$ (‘*Shubert penalized 2*’ function [4]). The two particles generate independent trajectories: the global minimum is detected.

TABLE II

TEST PROBLEMS: *In only 4 out of 35 cases, the random choice of the initial particles’ position for PSO is preferable, with respect to our proposal. Anyway, in these cases the difference is not dramatic as for many of the 31 remaining test functions.*

Problem	n	$f(x^*)_{independent}$	$f(x^*)_{random}$
Six humps camel back	2	-1.0316 *	-0.8118
Treccani	2	1.2125E-06 *	0.5983
Quartic	2	-0.1526 *	4.8516
Shubert	2	-49.2691 *	-37.5954
Shubert pen.1	2	-23.9748	-30.6048 *
Shubert pen.2	2	-23.2320 *	-18.8594
Shekel5	4	-3.36416 *	-0.4085
Shekel7	4	-3.39414 *	-0.5047
Shekel10	4	-1.30725 *	-0.9192
Exponential	2	-0.9999 *	-1.1805E-02
	4	-0.9999 *	-4.7240E-07
Cosine mixture	2	-0.2000 *	-4.6458E-02
Cosine mixture	4	-0.1042 *	9.1806E-02
hartman3	3	-3.6570 *	-0.9769
hartman6	6	-3.0208	-3.2795 *
5^n loc. minima	2	6.6128E-03 *	9.3625
	5	3.6934-02 *	5.1373E-02
	10	0.1103 *	1.1255
	20	0.2525 *	1.0001
	30	0.1453 *	0.7282
10^n loc. minima	2	2.81867E-02 *	45.9406
	5	1.2651 *	5.005
	10	2.3363 *	34.3466
	20	4.3714 *	20.5893
	30	3.3331 *	20.4258
15^n loc. minima	2	2.23199E-02 *	0.8580
	5	0.2709	3.9033E-02*
	10	0.8420 *	1.5038
	20	1.5152 *	5.4427
	30	2.6402	2.5683 *
Griewank	2	8.1694E-07 *	0.9505
	5	3.7250E-05 *	1.96844E-03
	10	3.8645E-05 *	0.5813
	20	1.8684E-05 *	0.6602
	30	8.5806E-05 *	0.5974