Chapter 1

GLOBALLY CONVERGENT MODIFICATIONS OF PARTICLE SWARM OPTIMIZATION FOR UNCONSTRAINED OPTIMIZATION

***Emilio F. Campana*[1]*, Giovanni Fasano*[1,2]*, Daniele Peri*[1∗]**

(1) Istituto Nazionale per Studi ed Esperienze di Architettura Navale (INSEAN), via di Vallerano, 139 – Roma, ITALY.

(2) Dipartimento di Matematica Applicata, Università Ca'Foscari Venezia, Dorsoduro 3825/e, Venezia, ITALY.

Key Words: Global Optimization, Unconstrained Optimization, Particle Swarm Optimization, Derivative-free methods, Globally Convergent methods.

AMS Subject Classification: 90C56, 90C59.

∗E-mail addresses: e.campana@insean.it, fasano@unive.it and d.peri@insean.it

1 Introduction

Most of the challenging applications of optimization in applied sciences involve a remarkably large computational cost, both in terms of machine resources and time of computation. This implies that the investigation and the use of optimization tools is a promising and active research area. The use and computability of the derivatives of a nonlinear function is just an example of the issues in this area (see for instance [6]).
In particular, engineering design encompasses real-world problems where both practitioners and theoreticians are continuously asked to provide robust solutions and theoretical advances. Indeed, here the large number of variables often requires hundreds or thousands of function evaluations (each evaluation possibly taking hours), in order to provide a reliable optimal design; not to mention the difficulties to generate and use the derivatives. Hence, the use of efficient and effective derivative-free methods seems an appealing topic to investigate, in order to solve the latter problems. Furthermore, in case the formulation of the problem in hand involves *noisy functions*, the use of the derivatives imposes strong limitations to adopt finite differences.

In this paper we focus on a modification of the PSO algorithm [12, 19], for the solution of the unconstrained *global optimization problem*

$$\min_{x \in \mathbb{R}^n} f(x), \qquad f : \mathbb{R}^n \to \mathbb{R}. \tag{1}$$

At present $f(x)$ is assumed to be a nonlinear and non-convex *continuous function*. More specifically, the PSO algorithm attempts at detecting a global minimum of (1), i.e. a point $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$, for any $x \in \mathbb{R}^n$.
When the function $f(x)$ is computationally costly, exact methods may be too expensive to solve (1), or they possibly do not provide a current satisfactory approximation of the solution. In these cases the use of heuristic methods may be fruitful, whenever the computational resources and/or the time allowed for the computation are severely limited. On this guideline, PSO proved to be both effective and efficient on several practical applications from real life [19].

On the other hand, the use of heuristic methods which also generate sequences of iterates satisfying convergence properties, would be much attractive. Thus, we focus here on some modifications of the PSO algorithm, where converging subsequences of iterates are generated. In particular, the modifications proposed guarantee that the generated sequences of iterates converge to stationary points of the objective function (see also [9, 10, 21, 23]). We carry on our analysis in order to provide sufficient conditions, which yield the convergence of the resulting method. Observe that the aim of this paper is to provide robust methods with a twofold purpose. First we want to gain advantage of PSO fast approach to a global solution; then, we aim at steering the solution towards a stationary point, satisfying first order optimality conditions of $f(x)$.
This paper is specifically concerned with theoretical properties of some modifications of PSO. A complete numerical experience on this issue requires a wide and detailed investigation, which is beyond the scope of the paper and deserves a paper on its own.

In this paper we use the subscripts to identify the particles in a PSO scheme, while the superscripts indicate the iteration. We denote by I the identity matrix of suitable dimension.

The symbol $w^k = O(z^k)$ indicates that there exist an index $\bar{k}$ and a constant $a > 0$, such that the real sequences $\{w^k\}$ and $\{z^k\}$ satisfy $|w^k| \leq a|z^k|$, for any $k > \bar{k}$.

In Section 2-3 we describe a generalized PSO iteration. Then, the Sections 4-5 introduce both the theory and the motivations for our modification of PSO iteration. Finally, in Sections 6-7 we describe some new algorithms and we carry out the related convergence analysis.

2 A generalized scheme for PSO

PSO solves (1) by iteratively generating subsequences of points in $\mathbb{R}^n$, which possibly approach a solution. At the current step of any subsequence, the next point both depends on the position of the current point in the subsequence, and the information of $f(x)$ provided by the other subsequences. In particular, we use the subscript j to indicate the subsequence, while the superscript k indicates the iterate in the subsequence.

We preliminarily consider the following PSO iteration for any $k \geq 0$ (see [1]):

$$\begin{aligned} v_j^{k+1} &= \chi \left[w^k v_j^k + c_j r_j \otimes (p_j^k - x_j^k) + c_g r_g \otimes (p_g^k - x_j^k) \right], \\ x_j^{k+1} &= x_j^k + v_j^{k+1}, \end{aligned} \tag{2}$$

where $j = 1, ..., P$ is used to indicate the j-th *particle* (i.e. the j-th subsequence of points), P is a finite integer, and the vectors v_j^k and x_j^k are n-real vectors, which respectively represent the *speed* (i.e. the search direction) and the *position* of the j-th particle at step k. With $r_j \otimes (p_j^k - x_j^k)$ (similarly with $r_g \otimes (p_g^k - x_j^k)$) we indicate that every entry of the vector $(p_j^k - x_j^k)$ must be multiplied by a different value of the coefficient r_j. Finally, the n-real vectors p_j^k and p_g^k satisfy the condition

$$\begin{aligned} &f(p_j^k) \leq f(x_j^\ell), \qquad \text{for any } \ell \leq k, \;\; p_j^k \in \{x_j^\ell\} \\ &f(p_g^k) \leq f(x_j^\ell), \qquad \text{for any } \ell \leq k \text{ and } j = 1, \ldots, P, \;\; p_g^k \in \{x_j^\ell\}, \end{aligned} \tag{3}$$

while $\chi, w^k, c_j, r_j, c_g, r_g$ are positive bounded coefficients. Observe that p_j^k represents the 'best position' in the j-th subsequence, while p_g^k is the 'best position' among all the subsequences. We recall that the choice of the coefficients is often problem dependent, though several standard values for them were proposed in the literature [3, 20, 27]. Anyway, general rules for assessing the coefficients in (2) are still under investigation.

Of course relations (2) include the case where either the *inertia* coefficient w^k or the *constriction* coefficient χ are used. Moreover, without loss of generality we assume that r_j and r_g are uniformly distributed random parameters with $r_j \in [0, 1]$ and $r_g \in [0, 1]$.

One possible generalization of (2) is obtained by assuming that possibly the speed v_j^{k+1} depends on the P vectors $p_h^k - x_j^k$ (see also [16]), $h = 1, \ldots, P$, and not only on the vectors $p_j^k - x_j^k$, $p_g^k - x_j^k$ [1]. The resulting new iteration for the j-th particle is given for any

$k = 0, 1, \ldots$ by

$$
\begin{aligned}
v_j^{k+1} &= \chi_j^k \left[w_j^k v_j^k + \sum_{h=1}^{P} c_{h,j} r_{h,j} (p_h^k - x_j^k) \right], \\
x_j^{k+1} &= x_j^k + v_j^{k+1}.
\end{aligned}
\tag{4}
$$

Observe that in (4) the coefficients $c_{h,j}$ and $r_{h,j}$ both depend on the particle j and the remaining particles (h). It can be readily seen [2] that for each particle j, assuming that $r_{h,j}$ is the same for all the entries of $(p_h^k - x_j^k)$, $\chi_j^k = \chi_j$ and $w_j^k = w_j$, for any $k \geq 0$, the iteration (4) is equivalent to the *discrete stationary (time-invariant) system*

$$
X_j(k+1) = \begin{pmatrix} \chi_j w_j I & -\sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} I \\ \chi_j w_j I & \left(1 - \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j}\right) I \end{pmatrix} X_j(k) + \begin{pmatrix} \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^k \\ \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^k \end{pmatrix},
\tag{5}
$$

where

$$
X_j(k) = \begin{pmatrix} v_j^k \\ x_j^k \end{pmatrix} \in \mathbb{R}^{2n}, \qquad k \geq 0.
\tag{6}
$$

We observe that the sequence $\{X_j(k)\}$ identifies the trajectory of the j-th particle in the state space $\mathbb{R}^{2n}$, and it can be split into the *free response* $X_{j\mathcal{L}}(k)$ and the *forced response* $X_{j\mathcal{F}}(k)$ (see also [18]). In other words, for any $k \geq 0$, $X_j(k)$ may be rewritten according with

$$
X_j(k) = X_{j\mathcal{L}}(k) + X_{j\mathcal{F}}(k),
\tag{7}
$$

where

$$
X_{j\mathcal{L}}(k) = \Phi_j(k) X_j(0), \qquad X_{j\mathcal{F}}(k) = \sum_{\tau=0}^{k-1} H_j(k-\tau) U_j(\tau),
\tag{8}
$$

and (after a few calculations -see also [1])

$$
\Phi_j(k) = \begin{pmatrix} \chi_j w_j I & -\sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} I \\ \chi_j w_j I & \left(1 - \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j}\right) I \end{pmatrix}^k,
\tag{9}
$$

$$
H_j(k-\tau) = \begin{pmatrix} \chi_j w_j I & -\sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} I \\ \chi_j w_j I & \left(1 - \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j}\right) I \end{pmatrix}^{k-\tau-1},
\tag{10}
$$

$$U_j(\tau) \quad = \quad \begin{pmatrix} \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^\tau \\ \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^\tau \end{pmatrix}. \tag{11}$$

We remark that unlike $X_{j\mathcal{F}}(k)$, the free response $X_{j\mathcal{L}}(k)$ in (7)-(8) only depends on the initial point $X_j(0)$, and not on the vector p_h^τ, $\tau \geq 0$. The latter observation will be largely used to carry out our results. In particular, the next section will be devoted to report some relevant analysis on the free response $X_{j\mathcal{L}}(k)$ (see also [3]).

3 Issues on the parameters assessment in PSO

It is well known (see for instance [18]) that if the j-th trajectory $\{X_j(k)\}$ in (7) is non-diverging, it satisfies the condition

$$\lim_{k\to\infty} X_j(k) = \lim_{k\to\infty} X_{j\mathcal{F}}(k), \qquad j = 1, \ldots, P;$$

i.e. the free response $X_{j\mathcal{L}}(k)$ is bounded away from zero only for finite values of the index k. Moreover, from (9) we have $\Phi_j(k) = \Phi_j(1)^k$, for any $k \geq 0$, and it was proved [2] that the $2n$ eigenvalues of the unsymmetric matrix $\Phi_j(1)$ are real. In particular, by setting for the sake of simplicity in (9)

$$a_j = \chi_j w_j, \qquad \omega_j = \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j}, \tag{12}$$

we can prove [1] that the matrix $\Phi_j(1)$ has at most the two distinct eigenvalues λ_{j1} and λ_{j2} with

$$\begin{aligned} \lambda_{j1} &= \frac{1 - \omega_j + a_j - \left[(1 - \omega_j + a_j)^2 - 4a_j\right]^{1/2}}{2} \\ \lambda_{j2} &= \frac{1 - \omega_j + a_j + \left[(1 - \omega_j + a_j)^2 - 4a_j\right]^{1/2}}{2}. \end{aligned} \tag{13}$$

In addition, each of them has algebraic multiplicity n. A necessary (but in general not sufficient) condition for the j-th trajectory $\{X_j(k)\}$ to be non-diverging, is provided by the following result (see also [18]), which imposes some *conditions on the coefficients of PSO iteration*.

Proposition 3.1 *Consider the PSO iteration (4). For any $j \in \{1, \ldots, P\}$ and any $k \geq 0$, let $r_{j,h}$ be the same for all the entries of the vector $(p_h^k - x_j^k)$, with $\chi_j^k = \chi_j$ and $w_j^k = w_j$. Suppose that for any particle $j \in \{1, \ldots, P\}$ the eigenvalues λ_{j1} and λ_{j2} in (13) satisfy the conditions*

$$\begin{aligned} |\lambda_{j1}| &< 1 \\ |\lambda_{j2}| &< 1. \end{aligned} \tag{14}$$

Then, for any j the sequence $\{X_{j\mathcal{L}}(k)\}$ satisfies $\lim_{k\to\infty} X_{j\mathcal{L}}(k) = 0$. The condition (14) is also a necessary condition for the trajectory $\{X_j(k)\}$ to be non-diverging. □

We highlight that most of the typical settings for PSO parameters proposed in the literature (see e.g. [3, 27]), satisfy the condition (14). In the light of the results in Proposition 3.1, a couple of issues still arise, which deserve further consideration.

1. The hypotheses in Proposition 3.1 neither ensure that for a fixed j the sequence $\{X_j(k)\}$ is converging, nor they guarantee that $\{X_j(k)\}$ admits limit points. I.e., for a fixed j there may be diverging subsequences of $\{X_j(k)\}$ even if (14) holds.

2. Suppose that the sequence $\{X_j(k)\}$ converges for $k \to \infty$, i.e. $\{X_j(k)\} \to X_j^*$ with (see (6))
$$X_j^* = \begin{pmatrix} v_j^* \\ \\ x_j^* \end{pmatrix}.$$
Then, x_j^* may fail to be a local minimum of $f(x)$, i.e. the property
$$f(x_j^*) \leq f(x), \qquad \forall x \text{ s.t. } \|x - x_j^*\| \leq \epsilon, \ \ \epsilon > 0,$$
may not be satisfied.

Observe that the first issue was addressed and partially investigated in [13, 21, 23]. Here we focus on the second issue above. On this purpose, we claim that under mild assumptions if the function $f(x)$ is continuously differentiable, it is possible to modify the PSO iteration in such a way that the sequence $\{x_1^1, \ldots, x_P^1, \ldots, x_1^k, \ldots, x_P^k\}$ admits *stationary limit points* for $f(x)$, i.e. either of the following properties holds
$$\begin{aligned} &\liminf_{k\to\infty} \left\| \nabla f(x_j^k) \right\| = 0 \\ &\lim_{k\to\infty} \left\| \nabla f(x_j^k) \right\| = 0. \end{aligned} \tag{15}$$

The next sections deal with the latter claim, which will be proved theoretically. We will give evidence that the satisfaction of condition (15) may be met at the expense of a reasonably larger computational cost, i.e. an increase of the number of function evaluations.

4 Our optimization framework

As described in the Introduction, in the last decades many design optimization and simulation-based optimization problems have claimed for more effective and robust methods, which do not *explicitly* use derivatives. Meanwhile, the advances on parallel and distributed computing have considerably helped to reduce the impact of the strong computational burden of challenging problems. The combination of the latter two trends has yielded a mature field of research, where efficient algorithms show both a complete convergence analysis and noteworthy performance: namely *direct search methods*. In the latter class we include (see [22, 13]) all the optimization methods which do not use derivatives and are simply based on “the ranks of a countable set of function values”.
In particular, we deal with iterative methods in the subclass of *Generating Set Search* (GSS),

where at each iteration a suitable set of search directions generating a cone is considered, in order to guarantee a decrease of the objective function. The *pattern search methods* [22] and the *derivative-free methods* [4, 7] are both included in GSS. The first provide convergence analysis by enforcing at each iteration a *simple decrease* of the objective function $f(x)$, on suitable geometric patterns. On the other hand, the second group imposes a *sufficient decrease* of the objective function by relying on a local model of $f(x)$. We highlight that the schemes described do not encompass several heuristics, which are often broadly used in the literature (see [13] and the cited references).
The local convergence analysis of GSS methods may be fruitfully combined with other techniques, in order to provide globally convergent algorithms. On this purpose, examples of combined methods where evolutionary strategies and GSS schemes yield globally convergent algorithms, can be found in [8, 9, 10, 24, 25]. In particular, in the last two references PSO is combined with a pattern search framework, in order to provide methods converging to stationary points.

Here, we similarly want to combine a PSO-based evolutionary scheme with a linesearch-based derivative-free algorithm, in order to provide a unified convergence analysis yielding (15). The latter approach is motivated by the promising performance of derivative-free methods when combined with a linesearch technique [14]. We also remark that a PSO-based approach combined with a trust-region framework was already proposed in the literature [24, 25], in order to provide methods converging to stationary points.

In this section we consider the solution of the problem (1), by means of a modified PSO scheme, combined with a derivative-free globally convergent algorithm, based on a linesearch strategy. We study in particular some convergence properties of the sequences $\{x_j^k\}$, $j = 1, \ldots, P$, under very mild assumptions on $f(x)$. Moreover, we propose four algorithms for continuously differentiable functions, whose distinguishing feature is the generation of sequences of points, which admit stationary limit points for $f(x)$. To the latter purpose, here we also impose additional conditions on the sequences of coefficients $\{\chi_j^k\}$, $\{w_j^k\}$, $\{c_{h,j}\}$, $\{r_{h,j}\}$ in (4).
We highlight that in accordance with formulae (7)-(8), here we impose in our analysis some *conservative* conditions on the forced response $X_{j\mathcal{F}}(k)$ of the particle j. In particular, in order to solve (1) with PSO, we have to guarantee that suitable PSO parameters exist such that the sequences $\{x_j^k\}$, for any j, admit limit points. On this guideline, as remarked above, the Proposition 3.1 provides only *necessary conditions* to guarantee the existance of limit points for the sequences $\{x_j^k\}$, $j = 1, \ldots, P$.

To sum up, in order to carry on a convergence analysis for PSO, in this section we focus on two main ingredients. First we provide conditions on the PSO iteration, in such a way that a bounded and closed set $\mathcal{L}_0$ exists which satisfies $\{x_j^k\} \subset \mathcal{L}_0$, for any j and k (so that for any fixed j the sequence $\{x_j^k\}$ admits limit points in $\mathcal{L}_0$). Then, recalling that PSO is a heuristics and therefore the sequence $\{p_g^k\}$ may not converge to a stationary point of $f(x)$ on $\mathcal{L}_0$, we modify as slightly as possible the PSO iteration so that either of the stationarity conditions holds

$$\liminf_{k\to\infty} \|\nabla f(p_g^k)\| = 0$$

$$\lim_{k\to\infty} \|\nabla f(p_g^k)\| = 0.$$

The resulting algorithms are modified PSO schemes, which guarantee that under mild assumptions at least a subsequence of the points $\{x_j^k\}$ converges to a stationary point (that is possibly a minimum point) of $f(x)$.

5 Preliminary theoretical results

We are concerned with analyzing the derivative-free approach in [14], which is based on a local model of the objective function. The proposals in [14] draw their inspiration from the idea of combining the pure *pattern search* and *derivative-free* approaches. Indeed, in [14] a suitable pattern of search directions is first identified, as in pattern search methods. Then, a one-dimensional linesearch is possibly performed along these directions, as in derivative-free schemes.
We report here some mild (*simplified*) conditions, which will be considered for generating search directions in modified PSO algorithms (see [14]).

Proposition 5.1 *Let $f : \mathbb{R}^n \to \mathbb{R}$, with $f \in C^1(\mathbb{R}^n)$. Suppose that for any k the points in the sequence $\{x^k\}$ are bounded. Let for any k the directions $\{d_j^k\}$, $j = 1, \ldots, n+1$, be* bounded *and satisfy one of the following two conditions:*

(a) the directions $d_1^k, \ldots, d_{n+1}^k$ form a positively spanning set of $\mathbb{R}^n$, i.e. for any $w \in \mathbb{R}^n$, there exist $n+1$ coefficients $\eta_j^k \geq 0$, $j = 1, \ldots, n+1$, such that $w = \sum_{j=1}^{n+1} \eta_j^k d_j^k$;

(b) the directions $d_1^k, \ldots, d_n^k$ are uniformly linearly independent. *Moreover, the bounded direction d_{n+1}^k satisfies*

$$d_{n+1}^k = \sum_{\ell=1}^{2n} \rho_\ell^k \left(\frac{w_1^k - w_\ell^k}{\xi_\ell^k} \right), \tag{16}$$

where

- *the sequences $\{\rho_\ell^k\}$, $\ell = 1, \ldots, 2n$, are* bounded*, with $\rho_\ell^k \geq 0$ and $\rho_{2n}^k \geq \rho > 0$, for all k;*
- *given the vectors $z_j^k \in \mathbb{R}^n$, $j = 1, \ldots, n$, for any k the vectors $\{w_1^k, \ldots, w_{2n}^k\}$ in (16) are defined by*

$$w_h^k = \begin{cases} z_{\lfloor h/2 \rfloor + 1}^k & h = 1, 3, 5, \ldots, 2n-1, \\ z_{h/2}^k + \xi_{h/2}^k d_{h/2}^k & h = 2, 4, 6, \ldots, 2n, \end{cases} \tag{17}$$

$$\xi_j^k > 0 \qquad j = 1, \ldots, n, \tag{18}$$

$$\lim_{k \to \infty} \xi_j^k = 0 \qquad j = 1, \ldots, n, \tag{19}$$

and

$*$ *the points $\{w_h^k\}$ are reordered and (possibly) relabelled in such a way that*

$$f(w_1^k) \leq f(w_2^k) \leq \cdots \leq f(w_{2n-1}^k) \leq f(w_{2n}^k);$$

* *there exist constants* $c_1, c_2 > 0$ *such that the sequences* $\{z_j^k\}$ *and* $\{\xi_j^k\}$ *satisfy*

$$\frac{\max\limits_{j=1,\ldots,n} \{\xi_j^k\}}{\min\limits_{j=1,\ldots,n} \{\xi_j^k\}} \leq c_1, \tag{20}$$

$$\|z_j^k - x^k\| \leq c_2 \xi_j^k, \qquad j = 1, \ldots, n; \tag{21}$$

* *the sequences* $\{\bar{\xi}_\ell^k\}$, $\ell = 1, \ldots, 2n$ *in (16), satisfy the condition*

$$\min_{j=1,\ldots,n} \{\xi_j^k\} \leq \bar{\xi}_\ell^k \leq \max_{j=1,\ldots,n} \{\xi_j^k\}. \tag{22}$$

Then, the following stationarity condition holds for the function $f(x)$

$$\lim_{k\to\infty} \|\nabla f(x^k)\| = 0 \qquad \Longleftrightarrow \qquad \lim_{k\to\infty} \sum_{j=1}^{n+1} \min\left\{0, \nabla f(x^k)^T d_j^k\right\} = 0. \tag{23}$$

□

Observe that considering the sequence $\{x^k\}$ in (23), Proposition 5.1 suggests that it is possible to provide necessary and sufficient conditions of stationarity. In particular, this can be accomplished by simply exploiting at any point of the sequence $\{x^k\}$ the objective function (through its directional derivatives), along the directions $d_1^k, \ldots, d_{n+1}^k$. Furthermore (see [14]), in Table 1 we report a derivative-free method for unconstrained minimization, which uses the results of Proposition 5.1, to generate sequences with stationary limit points. A full convergence analysis was developed for the Algorithm DF-0a and the following conclusion was proved (see [14] Proposition 5.1)

Proposition 5.2 *Suppose the directions* $d_1^k, \ldots, d_{n+1}^k$ *satisfy the Proposition 5.1. Consider the sequence* $\{x^k\}$ *generated by the Algorithm* DF-0a *and let the level set* $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ *be compact. Then we have*

$$\liminf_{k\to\infty} \|\nabla f(x^k)\| = 0. \tag{24}$$

□

Observe that the condition (24) is met only asymptotically; nevertheless, in the practical application of Algorithm DF-0a, a stopping condition occurs when $\bar{\alpha}^k$ at Steps 2 and 3 becomes too small. We also note that at Step 4 we can possibly choose $x^{k+1} \equiv y^k$, since the convergence analysis *does not* require $f(x^{k+1}) < f(y^k)$. Furthermore relation (24) may be consistently strengthened by adopting a different strategy at Step 2 of the Algorithm DF-0a. In particular, we highlight that at Step 2 just *one direction* of *sufficient decrease* for the objective function is sought. On the contrary, if we modify (reinforce) Step 2 and consider an exploitation of $f(x)$ along *all the directions* in the set $\{d_1^k, \ldots, d_{n+1}^k\}$, we obtain the Algorithm DF-0b in Table 2. The following proposition was also proved in [14] and provides a stronger result with respect to Proposition 5.2.

Step 0. Set $k = 0$; choose $x^0 \in \mathbb{R}^n$, set $\bar{\alpha}^0 > 0$, $\gamma > 0$, $\theta \in (0, 1)$.

Step 1. If there exists $y^k \in \mathbb{R}^n$ such that $f(y^k) \leq f(x^k) - \gamma\bar{\alpha}^k$, then go to **Step 4**.

Step 2. If there exists $j \in \{1, \dots, n+1\}$ and an $\alpha^k \geq \bar{\alpha}^k$ such that

$$f(x^k + \alpha^k d_j^k) \leq f(x^k) - \gamma(\alpha^k)^2,$$

then set $y^k = x^k + \alpha^k d_j^k$, set $\bar{\alpha}^{k+1} = \alpha^k$ and go to **Step 4**.

Step 3. Set $\bar{\alpha}^{k+1} = \theta\bar{\alpha}^k$ and $y^k = x^k$.

Step 4. Find x^{k+1} such that $f(x^{k+1}) \leq f(y^k)$, set $k = k + 1$ and go to **Step 1**.

Table 1. The derivative-free Algorithm DF-0a (see [14]).

Proposition 5.3 *Suppose the directions* $d_1^k, \dots, d_{n+1}^k$ *satisfy the Proposition 5.1. Consider the sequence* $\{x^k\}$ *generated by the Algorithm* DF-0b *and let the level set* $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ *be compact. Then we have*

$$\lim_{k \to \infty} \|\nabla f(x^k)\| = 0. \tag{25}$$

□

Observe that the stronger result is obtained at the cost of a more expensive Step 2, where the linesearch procedure and the cyclic use of *all* the directions $d_1^k, \dots, d_{n+1}^k$, may increase the number of function evaluations required to detect the stationary point. The use of the procedure `LINESEARCH()` is aimed to determine the smallest possible steplength α_j^k, such that a sufficient decrease of $f(x)$ is guaranteed.
We recall that the Propositions 5.2 and 5.3 provide only *local* convergence properties for the objective function $f(x)$, similarly to any gradient method for continuously differentiable functions. In other words, starting from any initial point $x^0 \in \mathbb{R}^n$, as long as the set $\mathcal{L}_0$ is compact, a stationary point which is possibly only a *local minimum* is asymptotically approached.

6 New algorithms

Now we want to couple the PSO scheme described in Section 2 with the algorithms in Tables 1 and 2, in order to possibly obtain new methods endowed with both *local* convergence properties and *global* strategies of exploration. In particular, we use the heuristic exploration of PSO to provide a *global* information on $f(x)$, then a derivative-free scheme is used to enforce the *local* convergence towards stationary points.

On this guideline, the most obvious way to couple PSO and derivative-free schemes is by performing the PSO iteration in Section 2 up to the finite iteration $\hat{k} \geq 0$. Then, we could apply either Algorithm DF-0a or Algorithm DF-0b after setting the initial point (see Tables 1 and 2) $x^0 = p_g^{\hat{k}}$. I.e., the local convergence is carried on starting from the best point detected by PSO up to step $\hat{k}$. Unfortunately, the latter strategy is a *blind* sequential application of two different algorithms, which does not join the advantages of the two approaches. On the contrary, we want to consider at once both the *exploitation* (the local strategy) and the *exploration* (the global strategy) of the objective function, at any step of a new scheme. More explicitly we consider a PSO-type scheme, which provides the investigation of a global minimum over $\mathbb{R}^n$, while retaining the asymptotic convergence properties of a (local) derivative-free technique.

We propose at first the Algorithm DF-1a in Table 3. It is a derivative-free method which uses for any iteration k the directions $d_1^k, \ldots, d_{n+1}^k$, described in Proposition 5.1. We can prove the following result (see also [14]).

Proposition 6.1 *Consider the Algorithm* DF-1a. *Suppose the directions* $d_1^k, \ldots, d_{n+1}^k$ *and the sequences* $\{z_j^k\}$, $j = 1, \ldots, n$, *satisfy the hypotheses of Proposition 5.1. Let the level set* $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$ *be compact. Then, the Algorithm* DF-1a *generates the sequence of points* $\{x^k\}$ *such that*

$$\liminf_{k \to \infty} \|\nabla f(x^k)\| = 0. \tag{26}$$

Proof.
Observe that the Algorithm DF-1a and the Algorithm DF-0a differ only at Step 1 and Step 4. In particular, observe that the Algorithm DF-1a is obtained from the Algorithm DF-0a, where the vectors y^k at Step 1 and x^{k+1} at Step 4 are computed by means of a PSO method. Thus, the convergence properties of the sequence generated by the Algorithm DF-1a follow straightforwardly from Proposition 5.2. □

We remark that the Steps 1 and 4 of the Algorithm DF-1a include the global exploration by using PSO. On the other hand the Steps 2 and 3 are substantially the same of the Algorithm DF-0a.

In a very similar fashion we can also couple the Algorithm DF-0b with a PSO method. Consequently, a conclusion as in Proposition 5.3 trivially holds for the resulting scheme (i.e. the condition (26) is reinforced giving condition (25)).

Another proposal to join PSO-type schemes and the linesearch-based derivative-free technique in Table 1, is the Algorithm DF-2a reported in Tables 4-5. At Step k the Algorithm DF-2a *exploits* the function $f(x)$ in a neighborhood of the point x^k, along the directions $d_1^k, \ldots, d_{n+1}^k$. Then, the new point x^{k+1} is generated in such a way that possibly a sufficient decrease of the objective function is obtained. Observe that at Step 0 of the Algorithm DF-2a, a suitable set of $n+1$ search directions is used, which meet the conditions of Proposition 5.1. In particular, according with the guidelines of Proposition 5.1, for any k we freely set the n uniformly linearly independent directions $d_1^k, \ldots, d_n^k$, which exploit *local information* on the function. Then, the direction d_{n+1}^k is generated by the resulting application of the modified PSO scheme in Table 5, which provides *global information* on the objective function. Observe that in particular, we apply here a PSO-based method with exactly n particles (nevertheless the results can be extended readily to the case of $P > n$ particles). Finally, we remark that at Step 4 of Algorithm DF-2a if $f(x^{k+1}) > f(y^k)$, then the PSO-based scheme (27)-(28) is substantially ineffective to improve the current iterate y^k.

7 How to generate search directions for global convergence

In order to prove the global convergence properties of the overall method in Tables 4-5, we remark that a modified PSO scheme must be adopted. In particular we focus on the algorithm in (27)-(28), which is supposed to include (at least) n particles. In this scheme we note that the position of the particles is (partially) affected by the choice of the directions $d_1^k, \dots, d_n^k$ (see relation (28)). Moreover, assuming that for any k the directions $d_1^k, \dots, d_n^k$ are assigned, in Table 5 we generate the direction d_{n+1}^k by applying the PSO scheme (27)-(28) and using the sequence $\{z_j^k\}$.
Observe that in formulae (27)-(28) the projection $P_{B(c,\rho)}(\cdot)$ onto the convex compact set $B(c, \rho)$ is introduced, with $c \in \mathbb{R}^n$, $\rho > 0$, in order to guarantee that the direction d_{n+1}^k is bounded. For any step k and any particle j we have (see also (2) and (4), where without loss of generality we have replaced '$\otimes$' with a simple multiplication)

$$\begin{cases} v_j^{k+1} = \chi_j^k \Bigg[w_j^k P_{B(c,\rho)}(v_j^k) + \displaystyle\sum_{h=1, h\neq g}^{n} c_{h,j}^k r_{h,j}^k \left[p_h^k - P_{B(c,\rho)}(z_j^k) \right] \\ \qquad\qquad + \; c_{g,j}^k r_{g,j}^k \left[x^{k+1} - P_{B(c,\rho)}(z_j^k) \right] \Bigg], \\ z_j^{k+1} = P_{B(c,\rho)}(z_j^k) + v_j^{k+1}, \end{cases} \tag{27}$$

where

$$\begin{aligned} & p_h^k = \mathrm{argmin}_{\ell \leq k} \left\{ f\left[P_{B(c,\rho)}(z_h^\ell) \right] \right\}, \qquad h = 1, \dots, n, \\ & x^{k+1} = \mathrm{argmin}_{\ell \leq k,\ h=1,\dots,n} \left\{ f\left[P_{B(c,\rho)}(z_h^\ell) \right], f\left[P_{B(c,\rho)}(z_h^\ell + \xi_h^\ell d_h^\ell) \right] \right\}, \end{aligned} \tag{28}$$

and $P_{B(c,\rho)}(y)$ indicates the *orthogonal projection* of vector $y \in \mathbb{R}^n$ onto the *compact* and *convex* set $B(c, \rho)$. Due to the computational burden which may be involved in the projection over a convex set, we suggest to choose $B(c, \rho)$ among the following possibilities (see Figure 1):

- $\rho \in \mathbb{R}$ and $B(c, \rho) = \{x \in \mathbb{R}^n : \|x - c\|_2 \leq \rho, \ \rho > 0\}$, which implies that for any $y \in \mathbb{R}^n$,

$$P_{B(c,\rho)}(y) = \begin{cases} y & \text{if } \|y - c\|_2 \leq \rho, \\ c + \rho \dfrac{y - c}{\|y - c\|_2} & \text{otherwise.} \end{cases}$$

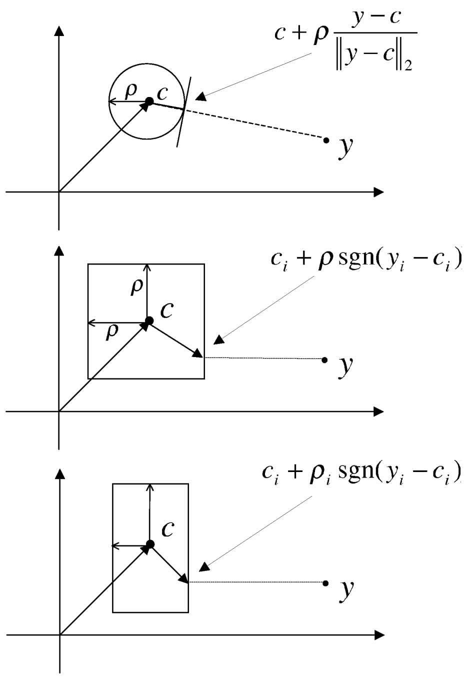

Figure 1. Projections over the compact convex set $B(c,\rho)$.

- $\rho \in \mathbb{R}$ and $B(c,\rho)=\{x\in\mathbb{R}^n : |x_i-c_i|\le\rho,\ i=1,\ldots,n,\ \rho>0\}$, which implies that for any $y\in\mathbb{R}^n$ and any $i=1,\ldots,n$

$$\left[P_{B(c,\rho)}(y)\right]_i = \begin{cases} y_i & \text{if } |y_i-c_i|\le\rho, \\ c_i+\rho\,\mathrm{sgn}(y_i-c_i) & \text{otherwise.} \end{cases}$$

- $\rho \in \mathbb{R}^n$ and $B(c,\rho)=\{x\in\mathbb{R}^n : |x_i-c_i|\le\rho_i,\ \rho_i>0,\ i=1,\ldots,n\}$, which implies that for any $y\in\mathbb{R}^n$ and any $i=1,\ldots,n$

$$\left[P_{B(c,\rho)}(y)\right]_i = \begin{cases} y_i & \text{if } |y_i-c_i|\le\rho_i, \\ c_i+\rho_i\,\mathrm{sgn}(y_i-c_i) & \text{otherwise.} \end{cases}$$

Now, let us consider the following assumption in order to prove the convergence results for the Algorithm DF-2a.

Assumption 7.1 *Consider the modified PSO scheme (27)-(28). Suppose the condition (14) holds. Let in (28) the coefficients* ξ_j^k*,* $j=1,\ldots,n$*,* $k\ge 0$*, be chosen as in Proposition 5.1. Let* $\mathcal{L}_0=\{x\in\mathbb{R}^n : f(x)\le f(x^0)\}$ *be compact and let the convex set* $B(c,\rho)$ *in (27)-(28) satisfy* $B(c,\rho)\supseteq\mathcal{L}_0$*. Assume that in iteration (27) the sequences* $\{\chi_j^k\}$*,* $\{w_j^k\}$*,* $\{c_{h,j}^k\}$*,* $\{r_{h,j}^k\}$ *satisfy*

(1) $\chi_j^k w_j^k = O(\xi_j^{k+1}),\qquad j = 1,\dots,n;$

(2) $\chi_j^k c_{h,j}^k r_{h,j}^k = O(\xi_j^{k+1}),\qquad j = 1,\dots,n,\ h = 1,\dots,n,\ h \neq g;$

(3) $\chi_j^k c_{g,j}^k r_{g,j}^k = 1 + O(\xi_j^{k+1}).\qquad j = 1,\dots,n.$

□

Note that the conditions (1), (2) and (3) in Assumption 7.1 can be readily fulfilled. Thus, the assumption on the coefficients $\{\xi_j^k\}$ is not particularly restrictive. On the other hand, also the assumption $B(c,\rho) \supseteq \mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$ is not strong, since our method is tailored for applications where physical bounds on the unknowns are usually easy to determine.
Now, we are ready to prove the following result, which ensures that under mild assumptions we can define a globally convergent modification of the PSO scheme in (27)-(28).

Proposition 7.1 *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and consider the Algorithm* DF-2a*. Let the level set $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$ be compact. Assume that for any step k the directions $d_1^k, \dots, d_n^k$ are uniformly linearly independent and bounded, and let the Assumption 7.1 hold. If the direction d_{n+1}^k in Algorithm* DF-2a *is generated by the procedure* PSO-gen$(\cdot)$*, then the Algorithm* DF-2a *generates the sequence $\{x_k\}$ such that*

$$(a) \qquad \{x^k\} \subset \mathcal{L}_0 \qquad \text{and} \qquad \liminf_{k\to\infty} \|\nabla f(x^k)\| = 0. \tag{29}$$

$$(b) \qquad \begin{cases} \liminf_{k\to\infty} \|z_j^k - x^k\| = 0, \ \ j = 1,\dots,n; \\ \\ \liminf_{k\to\infty} \|(z_j^k + \xi_j^k d_j^k) - x^k\| = 0, \ \ j = 1,\dots,n. \end{cases} \tag{30}$$

<u>**Proof**</u>
As regards (a), by the hypotheses, for any k the directions $d_1^k, \dots, d_n^k$ are bounded and uniformly linearly independent. Furthermore, consider the Proposition 5.1, along with the Step 0 and Step 4 of the Algorithm DF-2a. For any k, with the settings

$$\begin{array}{ll} c_1 = 1 & \\ \xi_j^k = \bar{\xi}_\ell^k = \dfrac{\beta_1}{(k+1)^{\beta_2}} & j = 1,\dots,n,\ \ell = 1,\dots,2n,\ \beta_1 > 0,\ \beta_2 > 0, \\ w_{2j}^k = z_j^k + \xi_j^k d_j^k & j = 1,\dots,n, \\ w_{2j-1}^k = z_j^k & j = 1,\dots,n, \\ w_1^k,\dots,w_{2n}^k & \text{are renamed and relabelled so that } f(w_1^k) \leq \cdots \leq f(w_{2n}^k), \\ \rho_\ell^k = 0 & \ell = 1,\dots,2n-1, \\ \rho_{2n}^k = 1 & \end{array}$$

the sequences $\{\xi_j^k\}$, $\{\bar{\xi}_l^k\}$ and $\{\rho_\ell^k\}$ satisfy (18), (19), (20), (22) of Proposition 5.1. Moreover, the vectors w_1^k, w_{2n}^k computed by the procedure PSO-gen$(\cdot)$ are bounded, inasmuch as by (27)-(28) z_j^k is bounded and d_j^k is bounded by the hypothesis, for any j. Now we prove that for any k both (21) holds and also the direction d_{n+1}^k, generated by the procedure

PSO-gen$(\cdot)$, is bounded. Thus, by the above choice of the coefficients ρ_ℓ^k, $\ell = 1,\dots,2n$, relation (16) becomes

$$d_{n+1}^k = \frac{1}{\bar{\xi}_{2n}^k}(w_1^k - w_{2n}^k).$$

Now, by the choice of ρ_ℓ^k and $\bar{\xi}_\ell^k$, for any index j the triangular inequality yields

$$\|d_{n+1}^k\| \leq \frac{\|w_1^k - w_{2n}^k\|}{\bar{\xi}_{2n}^k} = \frac{\|w_1^k - w_{2n}^k\|}{\xi_j^k} \leq \frac{\|w_1^k - x^k\|}{\xi_j^k} + \frac{\|w_{2n}^k - x^k\|}{\xi_j^k}, \tag{31}$$

where x^k is the current iterate in the sequence $\{x^k\}$, generated in (28) by the Algorithm DF-2a. Then, two cases have to be analyzed.
Either $w_1^k = z_j^k$ for some j (similarly if $w_{2n}^k = z_j^k$), or $w_1^k = z_j^k + \xi_j^k d_j^k$ for some j (similarly if $w_{2n}^k = z_j^k + \xi_j^k d_j^k$). In the first case we have from Assumption 7.1, formulae (27)-(28) and the boundedness of the convex set $B(c,\rho)$

$$\begin{aligned} \|w_1^k - x^k\| &= \|z_j^k - x^k\| \\ &= \Big\| P_{B(c,\rho)}(z_j^{k-1}) + O(\xi_j^k) P_{B(c,\rho)}(v_j^{k-1}) + \\ &\quad \sum_{h=1,h\neq g}^{n} O(\xi_j^k)\left[p_h^{k-1} - P_{B(c,\rho)}(z_j^{k-1})\right] + \qquad (32) \\ &\quad \left[1 + O(\xi_j^k)\right]\left[x^k - P_{B(c,\rho)}(z_j^{k-1})\right] - x^k \Big\| \\ &\leq \left\| P_{B(c,\rho)}(z_j^{k-1}) + O(\xi_j^k) + x^k - P_{B(c,\rho)}(z_j^{k-1}) - x^k \right\| \leq c_2 \xi_j^k \qquad (33) \end{aligned}$$

where $c_2 > 0$ and the first inequality follows from the relation

$$O(\xi_j^k) P_{B(c,\rho)}(v_j^{k-1}) + \sum_{h=1,h\neq g}^{n} O(\xi_j^k)\left[p_h^{k-1} - P_{B(c,\rho)}(z_j^{k-1})\right] + \\ + O(\xi_j^k)\left[x_k - P_{B(c,\rho)}(z_j^{k-1})\right] = O(\xi_j^k).$$

Otherwise, when $w_1^k = z_j^k + \xi_j^k d_j^k$ for some j (similarly if $w_{2n}^k = z_j^k + \xi_j^k d_j^k$), we have

$$\|w_1^k - x^k\| = \|z_j^k + \xi_j^k d_j^k - x^k\| = \left\|\left(z_j^k - x^k\right) + \xi_j^k d_j^k\right\| \leq c_2 \xi_j^k, \tag{34}$$

where $c_2 > 0$ and the last inequality follows from the boundedness of d_j^k and relation (33). Therefore, from (31)-(34) the direction d_{n+1}^k is bounded. Moreover, from (33)-(34) it is readily seen that for any j we have $\|z_j^k - x^k\| \leq c_2 \xi_j^k$, i.e. (21) holds.

Finally, by (28) the vector x^{k+1} is bounded for any k. Moreover, from the definition of x^0 we have $z_j^0 \in \mathcal{L}_0$, $j = 1,\dots,n$, and consequently from (28) $\{x^k\} \subset \mathcal{L}_0$. Indeed, either at Step 4 of Algorithm DF-2a the vector x^{k+1} is computed by (27)-(28), or it is $x^{k+1} = y^k$.

Since the directions $d_1^k,\dots,d_{n+1}^k$ satisfy Proposition 5.1, the results of Proposition 5.2 hold, i.e. the Algorithm DF-2a yields the condition (29).

As regards (b), the result follows directly by considering the relations (29) and (33)-(34). □

Remark 7.1 *A straightforward set of uniformly linearly independent directions* $d_1^k, \ldots, d_n^k$*, to be used in Proposition 7.1, is obtained by setting*

$$d_j^k = e_j, \qquad\qquad j = 1, \ldots, n,$$

where e_j *is the* j*-th unit vector. We highlight that for any* k *at Step 2 of the Algorithm* DF-2a*, the sufficient decrease of* $f(x)$ *is checked along the* $n+1$ *directions* $d_1^k, \ldots, d_{n+1}^k$*. Anyway, the first direction which satisfies the test is chosen and the cyclic check stops. Consequently, in order to use as frequently as possible the direction* d_{n+1}^k *(generated by the procedure* PSO-gen$(\cdot)$*), to update the point* y^k *at Step 2 of Algorithm* DF-2a*, the search over* $j \in \{1, \ldots, n+1\}$ *should preferably be started with* $j = n+1$*.*

Remark 7.2 *Observe that by (28) the computational cost per iteration* k *of the procedure* PSO-gen$(\cdot)$ *amounts to* $2n$ *function evaluations. Furthermore, item* (b) *of Proposition 7.1 shows that eventually all the particles will cluster around the stationary point detected.*

On the guidelines of Proposition 5.3, we aim at extending the results of Proposition 7.1 so that any subsequence of the sequence $\{x^k\}$ possibly converges to a stationary point. In particular, the Algorithm DF-2b in Table 6 meets the latter requirement and the following proposition holds.

Proposition 7.2 *Suppose* $f : \mathbb{R}^n \to \mathbb{R}$ *is continuously differentiable and consider the Algorithm* DF-2b*, let the level set* $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \le f(x^0)\}$ *be compact and Assumption 7.1 hold. Let for any step* k *the directions* $d_1^k, \ldots, d_n^k$ *be uniformly linearly independent and bounded. If the Algorithm* DF-2b *is applied, where* d_{n+1}^k *is generated by the procedure* PSO-gen*, then we have*

$$(a) \qquad \{x^k\} \subset \mathcal{L}_0 \qquad \text{and} \qquad \lim_{k\to\infty} \|\nabla f(x^k)\| = 0. \tag{35}$$

$$(b) \qquad \begin{cases} \lim_{k\to\infty} \|z_j^k - x^k\| = 0, \;\; j = 1, \ldots, n \\ \\ \lim_{k\to\infty} \|(z_j^k + \xi_j^k d_j^k) - x^k\| = 0, \;\; j = 1, \ldots, n \end{cases} \tag{36}$$

Proof
The proof trivially follows by observing the correspondence of the Algorithms DF-0a and DF-0b, with the Algorithms DF-2a and DF-2b. Thus, the results of Propositions 5.3 and 7.1 yield (35)-(36). □

8 Conclusions

In this paper we have considered four different globally convergent modifications of the PSO iteration, applied for the solution of unconstrained global optimization problems. We have proved in Propositions 6.1 and 7.1 that under mild assumptions, at least a subsequence of the iterates produced by our modified PSO methods converges to a stationary point, which is possibly a minimum point. This is a relatively strong result, if we consider that by no means the standard PSO iteration [12] can guarantee the convergence towards stationary points. In addition, the latter result is in our knowledge among the first schemes (see also [24, 25]) where a modified PSO scheme is proved to be globally convergent, i.e. either (29) or (35) holds. Moreover, this accomplishment contributes to fill the gap between the theory and the numerical performance of PSO based methods.

Our conclusions have also been reinforced in Proposition 7.2, where *any* subsequence generated by the modified PSO iteration was proved to converge to a stationary point. We highlight that this stronger result implies an additional computational burden. However, the latter additional cost may be consistently reduced according with the indication reported in the Remark 7.1.

Finally, considering the wide range of applications which require the use of efficient derivative-free algorithms, we guess that a new paper will be necessary to describe further theoretical results, along with the numerical tests. In particular, the removal of the *continuous differentiability* assumption for the objective function $f(x)$, seems the natural extension of the theory described here. On this purpose, we need to include in our approach several results from non-smooth analysis.
On the other hand, the extension of our approach to bounded and linearly constrained problems, is another topic of great interest.

Acknowledgements

E.F.Campana and D.Peri would also like to thank the support of the US Office of Naval Research through Dr. Ki-Han KIM (NICOP project 00014-0810957). G.Fasano wishes to thank the INSEAN research program "*VISIR*", and Programma PRIN 20079PLLN7 "*Nonlinear Optimization, Variational Inequalities, and Equilibrium Problems*".

References

[1] E.F. Campana, G. Fasano, A. Pinto, Dynamic analysis for the selection of parameters and initial population, in Particle Swarm Optimization, Accepted for publication on Journal of Global Optimization, 2009.

[2] E.F. Campana, G. Fasano, A. Pinto, Dynamic system analysis and initial particles position in Particle Swarm Optimization, IEEE Swarm Intelligence Symposium 2006, Indianapolis 12-14 May 2006.

[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] R. De Leone, M. Gaudioso, and L. Grippo, Stopping criteria for linesearch methods without derivatives, Mathematical Programming, No. 30, pp. 285-300, 1984.

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

[7] U.M. Garcia-Palomares and J.F. Rodriguez, New sequential and parallel derivative-free algorithms for unconstrained minimization, SIAM Journal of Optimization, No. 13, pp. 79-96.

[8] W.E. Hart, Evolutionary Pattern Search Algorithms, Technical Report SAND95-2293, Sandia National Laboratories, Albuquerque, NM, 1995.

[9] W.E. Hart, A stationary point convergence theory for evolutionary algorithms, Foundations of Genetic Algorithms 4, Morgan Kaufmann, San Francisco, CA, 1996, pp. 325-342.

[10] W.E. Hart, A generalized stationary point convergence theory for evolutionary algorithms, Proceedings of the International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, 1997, pp. 127-134.

[11] J. Haslinger, R.A.E. Mäkinen Introduction to Shape Optimization, SIAM Advances in Design and Control, Philadelphia, 2003.

[12] 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.

[13] T.G. Kolda, R.M. Lewis, V. Torczon, Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods SIAM Review Vol. 45, No. 3, pp. 385-482.

[14] S. Lucidi, M. Sciandrone, On the global convergence of derivative-free methods for unconstrained optimization, SIAM Journal of Optimization, No. 13, pp. 97-116, 2002.

[15] B. Mohammadi, O. Pironneau, Applied Shape Optimization for Fluids, Clarendon Press, Oxford, 2001.

[16] R. Mendes, Population Topologies and Their Influence in Particle Swarm Performance, PhD Dissertation, University of Minho, Departamento de Informtica Escola de Engenharia Universidade do Minho, 2004.

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

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

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

[20] J.F. Schutte, A.A. Groenwold, A Study of Global Optimization Using particle Swarms, Journal of Global Optimization, No. 31, pp. 93-108, 2005.

[21] I. C. Trelea, The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection Information Processing Letters, vol. 85, pp. 317-325, 2003.

[22] M.W. Trosset I know it when I see it: Toward a definition of direct search methods, SIAG/OPT Views-and-News: A Forum for the SIAM Activity Group on Optimization, No. 9, pp. 7-10, 1997.

[23] F. Van den Bergh, A. P. Engelbrecht, A Study of Particle Swarm Optimization Particle Trajectories Information Sciences, vol. 176, pp. 937-971, 2006.

[24] A. I. F. Vaz, L. N. Vicente, A particle swarm pattern search method for bound constrained global optimization Journal of Global Optimization, vol. 39, pp. 197-219, 2007.

[25] A. I. F. Vaz, L. N. Vicente, PSwarm: A hybrid solver for linearly constrained global derivative-free optimization Optimization Methods and Software, vol. 24, pp. 669-685, 2009.

[26] 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.

[27] Y.L. Zheng, L.H. Ma, L.Y. Zhang, J.X. Qian, On 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.

Step 0. Set $k = 0$. Choose $x^0 \in \mathbb{R}^n$ and $\bar{\alpha}_j^0 > 0$, $j = 1, \ldots, n+1$, $\gamma > 0$, $\delta \in (0,1)$, $\theta \in (0,1)$.

Step 1. Set $j = 1$ and $y_1^k = x^k$.

Step 2. If $f(y_j^k + \bar{\alpha}_j^k d_j^k) \leq f(y_j^k) - \gamma(\bar{\alpha}_j^k)^2$ then

compute α_j^k by LINESEARCH$(\bar{\alpha}_j^k, y_j^k, d_j^k, \gamma, \delta)$ and set $\bar{\alpha}_j^{k+1} = \alpha_j^k$;

else set $\alpha_j^k = 0$, and $\bar{\alpha}_j^{k+1} = \theta \bar{\alpha}_j^k$.

Set $y_{j+1}^k = y_j^k + \alpha_j^k d_j^k$.

Step 3. If $j < n+1$ then set $j = j+1$ and go to **Step 2**.

Step 4. Find x^{k+1} such that $f(x^{k+1}) \leq f(y_{n+1}^k)$, set $k = k+1$ and go to **Step 1**.

LINESEARCH$(\bar{\alpha}_j^k, y_j^k, d_j^k, \gamma, \delta)$:

Compute the steplength $\alpha_j^k = \min\left\{\bar{\alpha}_j^k/\delta^h,\ h = 0, 1, \ldots\right\}$ such that

$$f(y_j^k + \alpha_j^k d_j^k) \leq f(x^k) - \gamma(\alpha_j^k)^2,$$

$$f\left(y_j^k + \frac{\alpha_j^k}{\delta} d_j^k\right) \geq \max\left[f(y_j^k + \alpha_j^k d_j^k), f(y_j^k) - \gamma\left(\frac{\alpha_j^k}{\delta}\right)^2\right].$$

Table 2. The derivative-free Algorithm DF-0b in [14].

Data. Set $k = 0$; choose $x^0 \in \mathbb{R}^n$ and $z_j^0, v_j^0 \in \mathbb{R}^n$, $j = 1, \dots, P$. Set $\bar{\alpha}^0 > 0$, $\gamma > 0$, $\theta \in (0, 1)$.

Step 1. Set $h_k \geq 1$ integer. Apply h_k PSO iterations considering the P particles with respective initial velocities and positions v_j^k and z_j^k, $j = 1, \dots, P$. Set $y^k = argmin_{1 \leq j \leq P,\ \ell \leq h_k} \{f(z_j^\ell)\}$. If $f(y^k) \leq f(x^k) - \gamma \bar{\alpha}^k$, then set $v_j^k = v_j^{k+h_k}$ and $z_j^k = z_j^{k+h_k}$, and go to **Step 4**.

Step 2. If there exists $j \in \{1, \dots, n+1\}$ and an $\alpha^k \geq \bar{\alpha}^k$ such that

$$f(x^k + \alpha^k d_j^k) \leq f(x^k) - \gamma(\alpha^k)^2,$$

then set $y^k = x^k + \alpha^k d_j^k$, $\bar{\alpha}^{k+1} = \alpha^k$ and go to **Step 4**.

Step 3. Set $\bar{\alpha}^{k+1} = \theta \bar{\alpha}^k$ and $y^k = x^k$.

Step 4. Set $q_k \geq 1$ integer. Apply q_k PSO iterations considering the P particles with respective initial velocities and positions v_j^k and z_j^k, $j = 1, \dots, P$. Set $x^{k+1} = argmin_{1 \leq j \leq P,\ \ell \leq q_k} \{f(z_j^\ell)\}$; if x^{k+1} satisfies $f(x^{k+1}) \leq f(y^k)$, then set $k = k + 1$ and go to **Step 1**.

Table 3. The derivative-free Algorithm DF-1a.

Data. Set $k = 0$, choose $z_j^0 \in \mathbb{R}^n$, $j = 1, \ldots, n$. Set $x^0 = argmax_{1 \le j \le n}\{f(z_j^0)\}$.
Let $\bar{\alpha}^0 > 0$, $\beta_1 > 0$, $\beta_2 > 0$, $\gamma_1 > 0$, $c_2 > 0$, $\theta \in (0, 1)$.

Step 0. Set $\xi_1^k = \cdots = \xi_n^k = \beta_1/(k+1)^{\beta_2}$. Either set $d_1^k, \ldots, d_{n+1}^k$ as in (b) of Proposition 5.1, or compute $d_1^k, \ldots, d_n^k$ as in (b) of Proposition 5.1 and d_{n+1}^k by using procedure PSO-gen$(k; d_1^k, \ldots, d_n^k; z_1^k, \ldots, z_n^k; \xi_1^k, \ldots, \xi_n^k)$.

Step 1. If there exists $y^k \in \mathbb{R}^n$ such that $f(y^k) \le f(x^k) - \gamma_1 \bar{\alpha}^k$, then go to **Step 4**.

Step 2. If there exists $j \in \{1, \ldots, n+1\}$ and an $\alpha^k \ge \bar{\alpha}^k$ such that

$$f(x^k + \alpha^k d_j^k) \le f(x^k) - \gamma_1 (\alpha^k)^2,$$

then set $y^k = x^k + \alpha^k d_j^k$, $\bar{\alpha}^{k+1} = \alpha^k$ and go to **Step 4**.

Step 3. Set $\bar{\alpha}^{k+1} = \theta \bar{\alpha}^k$ and $y^k = x^k$.

Step 4. Let x^{k+1} and z_j^{k+1}, $j = 1, \ldots, n$, (possibly) satisfy (27)-(28).
If $f(x^{k+1}) \le f(y^k)$ then go to **Step 0**, else set $x^{k+1} = y^k$ and choose $z_j^{k+1} \in \mathbb{R}^n$, $j = 1, \ldots, n$. Set $k = k + 1$, and go to **Step 0**.

Table 4. The derivative-free Algorithm DF-2a.

Data: $k; d_1^k, \ldots, d_n^k; z_1^k, \ldots, z_n^k; \xi_1^k, \ldots, \xi_n^k; \rho > 0, \beta_1 > 0, \beta_2 > 0, c \in \mathbb{R}^n$.

Step k: Compute the vectors w_1^k and w_{2n}^k as

$$w_1^k = \mathrm{argmin}_{1 \le j \le n} \left\{ f(z_j^k), f(z_j^k + \xi_j^k d_j^k) \right\},$$

$$w_{2n}^k = \mathrm{argmax}_{1 \le j \le n} \left\{ f(z_j^k), f(z_j^k + \xi_j^k d_j^k) \right\}.$$

Compute the direction d_{n+1}^k as

$$d_{n+1}^k = \frac{(k+1)^{\beta_2}}{\beta_1}(w_1^k - w_{2n}^k).$$

Table 5. The procedure PSO-gen$(k; d_1^k, \ldots, d_n^k; z_1^k, \ldots, z_n^k; \xi_1^k, \ldots, \xi_n^k)$.

Data. Set $k=0$, choose $z_j^0 \in \mathbb{R}^n$, set $x^0 = argmax_{1 \le j \le n}\{f(z_j^0)\}$, $j=1,\ldots,n$.
Let $\bar{\alpha}^0 > 0$, $j=1,\ldots,n$, $\beta_1 > 0$, $\beta_2 > 0$, $\gamma > 0$, $c_2 > 0$, $\theta \in (0,1)$, $\delta \in (0,1)$.

Step 0. Set $\xi_1^k = \cdots = \xi_n^k = \beta_1/(k+1)^{\beta_2}$. Either set $d_1^k,\ldots,d_{n+1}^k$ as in (b) of Proposition 5.1, or compute $d_1^k,\ldots,d_n^k$ as in (b) of Proposition 5.1 and d_{n+1}^k by means of the procedure PSO-gen$(k; d_1^k,\ldots,d_n^k; z_1^k,\ldots,z_n^k; \xi_1^k,\ldots,\xi_n^k)$.

Step 1. Set $j=1$ and $y_1^k = x^k$.

Step 2. If $f(y_j^k + \bar{\alpha}_j^k d_j^k) \le f(y_j^k) - \gamma(\bar{\alpha}_j^k)^2$ then

compute α_j^k by LINESEARCH$(\bar{\alpha}_j^k, y_j^k, d_j^k, \gamma, \delta)$ and set $\bar{\alpha}_j^{k+1} = \alpha_j^k$;

else set $\alpha_j^k = 0$, and $\bar{\alpha}_j^{k+1} = \theta\bar{\alpha}_j^k$. Set $y_{j+1}^k = y_j^k + \alpha_j^k d_j^k$.

Step 3. If $j < n+1$ set $j = j+1$ and go to **Step 2**.

Step 4. Let x^{k+1} and z_j^{k+1}, $j=1,\ldots,n$, (possibly) satisfy (27)-(28).
If $f(x^{k+1}) \le f(y_{n+1}^k)$ then go to **Step 0**, else set $x^{k+1} = y_{n+1}^k$ and choose $z_j^{k+1} \in \mathbb{R}^n$, $j=1,\ldots,n$. Set $k=k+1$, and go to **Step 0**.

Table 6. The derivative-free Algorithm DF-2b.