Università
Ca'Foscari
Venezia

**Department
of Management**

Working Paper Series

**M. Corazza, G. Di Tollo, G. Fasano
and R. Pesenti**

A novel initialization of PSO for costly portfolio selection problems

**Working Paper n. 4/2015
July 2015**

ISSN: 2239-2734

A novel initialization of PSO for costly portfolio selection problems

Marco Corazza[1], Giacomo Di Tollo[2], Giovanni Fasano[2, 3] and Raffaele Pesenti[2]

(1) Department of Economics, Ca' Foscari University of Venice,
Sestiere di Cannaregio 873, 30121 Venezia, Italy,
E-mail: corazza@unive.it.

(2) Department of Management, Ca' Foscari University of Venice,
Sestiere di Cannaregio 873, 30121 Venezia, Italy,
E-mail: {giacomo.ditollo, fasano, pesenti}@unive.it.

(2) National Research Council - Maritime Research Centre (CNR-INSEAN),
Via di Vallerano 139, 00128 Roma, Italy.

Abstract

In this paper we propose an efficient initialization of a deterministic Particle Swarm Optimization (PSO) scheme. PSO has showed to be promising for solving several unconstrained global optimization problems from real applications, where derivatives are unavailable and the evaluation of the objective function tends to be costly. Here we provide a theoretical framework which motivates the use of a deterministic version of PSO, in place of the standard stochastic iteration currently adopted in the literature. Then, in order to test our proposal, we include a numerical experience using a realistic complex portfolio selection problem. This numerical experience includes the application of PSO to a parameter dependent unconstrained reformulation of the considered portfolio selection problem. The parameters are either adaptively updated as in an exact penalty framework, or they are tuned by the code REVAC. We show that in both these settings our PSO initialization is preferable with respect to the standard proposal from the literature.

Keywords: Deterministic PSO, Global Optimization, Portfolio Selection Problems, Exact Penalty functions.

1 Introduction

Particle Swarm Optimization (PSO) is a well known bio-inspired population-based metaheuristic for unconstrained global optimization problems. In this metaheuristic, a crucial role is played by the random initialization of some parameters (namely, the positions and the velocities of the swarm members), that can affect the final solution. In this paper we propose a novel deterministic method for initializing these parameters. In particular, we prove that, under mild analytical assumptions, the proposed initialization permits to the swarm particles to scout the search space following nearly orthogonal directions (the technical way in which we mean "orthogonal" is clarified in Section 4). This guarantees that the particles can be spread more homogeneously in the search space, so that we reduce the possibility of having unexplored subsets containing potential solutions (see also [1]). All that is not ensured by the random initialization.

Then, in order to test our deterministic initialization, we perform a numerical experience using as optimization problem the unconstrained reformulation of a realistic complex portfolio selection problem. In particular, the considered problem is the one proposed in [2], which is nonconvex, nondifferentiable, mixed-integer and NP-hard (in this sense we mean that this portfolio selection problem is "complex"). In this regard, it is noted that the portfolio management industry usually does not possess the mathematical knowledge and/or the research capabilities to handle such a type of problems. Furthermore, it could be not convenient for it to set up a team of external experts. As a consequence, part of the investors' demand might remain unsatisfied or satisfied by the use of inappropriate solution technologies. Therefore, PSO improved by our novel deterministic initialization can represent a flexible operational tool for the fund management industry, when searching in real time for approximate optimal portfolios associated with different economic scenarios. In other terms, in this industrial context the use of fast global approximate solvers providing robust solutions is preferable to resource-consuming exact optimizers.

With regards to the unconstrained reformulation of the complex portfolio selection problem, we adopt an exact penalty method (see also [2] and the references therein). Of course, other possible reformulations of the mathematical programming problem may be considered. Anyway, here we do not investigate this aspect. Furthermore, it is noteworthy that, generally speaking, the unconstrained reformulation of a constrained optimization problem has to take into account a pair of distinct goals. In fact, the chosen solution methodology for the reformulation has simultaneously to manage both the optimization of the original objective function, and the minimization of all constraints violations. On this guideline, in this paper we test PSO using our novel deterministic initialization, over two unconstrained reformulations of portfolio selection problems. In the first one, penalty parameters are adaptively updated in an exact penalty framework, and in the second one they are assessed through the tuning procedure REVAC [3]. The results from the numerical experience show that, adopting either of the above reformulations, our initialization seems to large extent preferable with respect to the standard one proposed in the literature. In particular, we claim that the latter fact holds when a reduced number of PSO iterations is allowed (e.g. for problems where the objective function is costly). Indeed, we provide both a theoretical motivation and a numerical evidence that in the early iterations of PSO our proposal can show a faster progress.

The remainder of this paper is organized as follows. In the next section we recall the basics of PSO. In Section 3 we highlight the impact on the search process of the initialization of the positions and of the velocities of the swarm particles. In Section 4 we propose our novel deterministic method which yields the latter initialization. In Section 5 we briefly present the complex portfolio selection problem we use for checking our deterministic initialization. In Section 6 we describe the plan of our experience, along with the two variants developed for the setting of the parameters characterizing the unconstrained reformulation of the portfolio problems. In Section 7 we report the main results and statistics obtained by applying PSO, improved by our deterministic initialization. In Section 8 we provide some basics on exact penalty functions and we motivate their use in our framework. In Section 9 we reformulate the complex portfolio selection problem as a nonlinear mixed-integer programming problem, in order to exactly solve it and to compare the resulting solutions with the one obtained through our PSO method. Some final remarks are given in Section 10.

As regards the notation adopted in this paper, for the sake of clarity we use *greek* letters like ξ and ν to denote vectors related to PSO particles, while *latin* letters like x and z will be used to indicate quantities in the reformulations of portfolio problem. Finally, for the sake of clarity and to avoid possible confusion, the i-th entry of the vector s might be denoted by either s_i or $(s)_i$.

2 Basics on PSO

As in the landmark paper [4], PSO is a bio-inspired optimization methodology that iteratively tries to improve a set of candidate solutions, the so-called *particles*, in global optimization problems. Several variants were proposed in the literature, both for unconstrained [5] and constrained [6, 7, 8] problems.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be the function to minimize, and for a given point $\bar{y} \in \mathbb{R}^n$ let the level set

$$\mathcal{L}_f(\bar{y}) = \{y \in \mathbb{R}^n \ : \ f(y) \leq f(\bar{y})\}$$

be compact. Then, the basic PSO iteration at step $k \geq 0$, for the particle (i.e. candidate solution) j is

$$\xi_j^{k+1} = \xi_j^k + \nu_j^{k+1}, \qquad j = 1, \ldots, P, \tag{1}$$

where $\xi_j^k \in \mathbb{R}^n$ is the current position of the j-th particle in the population, while $\nu_j^{k+1} \in \mathbb{R}^n$ is the current velocity of particle j (i.e., a *search direction* in the language of optimization). Thus, ξ_j^{k+1} will be the new position of particle j in the next step.

Unlike the standard gradient based methods, the vector ν_j^{k+1} is not necessarily a descent direction for function $f(y)$ at ξ_j^k. This suggests that at step k the j-th particle might not provide a new iterate which improves the objective function value. The direction ν_j^{k+1} is a cone combination of three contributions: the vector ν_j^k (which represents the so called *inertia* of particle j to change trajectory), a vector which uniquely depends on the previous iterates of particle j, and a vector which uniquely depends on the progress of the entire swarm, up to step k. In particular, we have for v_j^{k+1} the expression

$$\nu_j^{k+1} = \nu_j^k + \alpha_j^k \otimes (p_j^k - \xi_j^k) + \beta_j^k \otimes (p_g^k - \xi_j^k), \qquad k \geq 0, \tag{2}$$

where $\alpha_j^k, \beta_j^k \in \mathbb{R}^n$ are positive random vectors, the symbol '$\otimes$' indicates the entry-by-entry product between vectors, and the vector p_j^k, respectively p_g^k, is the best solution so far found respectively by particle j and by the swarm:

$$p_j^k = \arg \min_{0 \leq h \leq k} \left\{ f(\xi_j^h) \right\}, \qquad j = 1, \ldots, P, \tag{3}$$

$$p_g^k = \arg \min_{\substack{0 \leq h \leq k \\ j=1,\ldots,P}} \left\{ f(\xi_j^h) \right\}. \tag{4}$$

In (2), the parameters α_j^k (*cognitive parameter*) and β_j^k (*social parameter*), respectively consider the contribution to ν_j^{k+1} from the history of the j-th particle (i.e. $p_j^k - \xi_j^k$), and from the history of the entire swarm (i.e. $p_g^k - \xi_j^k$). Their general expressions in the literature are

$$\alpha_j^k = c_j^k r_1^k, \qquad r_1^k \in U[0,1]^n, \qquad k = 0, 1, \ldots,$$

$$\beta_j^k = c_g^k r_2^k, \qquad r_2^k \in U[0,1]^n, \qquad k = 0, 1, \ldots,$$

being $U[0,1]^n$ the uniform distribution with n entries between 0 and 1, and typically $c_j^k, c_g^k \in (0, 2.5]$.

In this paper, we will consider the following slightly more general reformulation of PSO iteration of each particle $j \in \{1, \ldots, P\}$ at any step $k \geq 0$ as proposed in [9]:

$$\begin{cases} \nu_j^{k+1} = \chi^k \left[w^k \nu_j^k + \alpha_j^k \otimes (p_j^k - \xi_j^k) + \beta_j^k \otimes (p_g^k - \xi_j^k) \right], & k \geq 0, \\ \\ \xi_j^{k+1} = \xi_j^k + \nu_j^{k+1}, & k \geq 0. \end{cases} \tag{5}$$

where χ^k is known as the *constriction coefficient*, and w^k as the *inertia coefficient*.

3 The impact of initialization on PSO

This section highlights the impact of the the choice of the initial points

$$\begin{cases} \xi_j^0 \in \mathbb{R}^n, & j = 1, \dots, P, \\ \nu_j^0 \in \mathbb{R}^n, & j = 1, \dots, P, \end{cases}$$

in the evolution of the positions and velocities of PSO particles. To this purpose we consider relation (5), where the parameters are chosen according with the following assumptions.

Assumptions 1 *In (5) we assume that the level set $\mathcal{L}_f(\bar{y})$, with $\bar{y} = \arg\min_{1 \le j \le P}\{f(\xi_j^0)\}$ is compact, and that*

$$\begin{aligned}
& c_j^k = c > 0, && \forall k \ge 0, \quad \forall j \in \{1, ..., P\}, && \text{(6a)}\\
& c_g^k = c_g > 0, && \forall k \ge 0, && \text{(6b)}\\
& r_1^k = r\mathbf{1} > 0, && \forall k \ge 0, && \text{(6c)}\\
& r_2^k = r_g\mathbf{1} > 0, && \forall k \ge 0, && \text{(6d)}\\
& w^k = w > 0, && \forall k \ge 0, && \text{(6e)}\\
& \chi^k = \chi > 0, && \forall k \ge 0, && \text{(6f)}\\
& \chi w < 1, && && \text{(6g)}\\
& \chi(cr + c_g r_g) < 2(\chi w + 1), && && \text{(6h)}\\
& \chi(cr + c_g r_g) \ne (1 \pm \sqrt{\chi w})^2, && && \text{(6i)}
\end{aligned}$$

where c, c_g, r, r_g, w, χ are suitable positive bounded coefficients, and $\mathbf{1} = (1, \cdots, 1)^T \in \mathbb{R}^n$. That is, the parameters $\{c_j^k\}$, $\{c_g^k\}$, $\{r_1^k\}$, $\{r_2^k\}$, $\{w^k\}$ and $\{\chi^k\}$ are invariant with respect to the particles and the iteration step.

Assumptions (6a), (6b), (6e) and (6f) are quite common in PSO literature [9]. Assumptions (6g) and (6h) are necessary (but possibly non sufficient) to guarantee the convergence of PSO, as we will discuss later in this section. Almost all the classic choices for the coefficients in PSO comply with these assumptions [10]. Assumption (6i) is not really necessary, but will simplify the arguments later in this section by only minimally reducing their generality. Finally, assumptions (6c) and (6d) imply that the PSO iterations are *deterministic* as the values of r and r_g are not stochastic. Motivations for deterministic PSO were given in the Introduction, and will no more be analyzed here.

In the light of the above assumptions, we can rewrite (5) as

$$\begin{cases} \nu_j^{k+1} = \chi\left[w\nu_j^k + cr(p_j^k - \xi_j^k) + c_g r_g(p_g^k - \xi_j^k)\right], & k \ge 0, \\ \xi_j^{k+1} = \xi_j^k + \nu_j^{k+1}, & k \ge 0. \end{cases} \tag{7}$$

Let us denote by I the identity matrix in $\mathbb{R}^{n \times n}$. Then, we can define the vector

$$X_j(k) = \begin{pmatrix} \nu_j^k \\ \xi_j^k \end{pmatrix} \in \mathbb{R}^{2n}$$

as *state* of the particle j at iteration k, whose components are the velocity ν_j^k and the position ξ_j^k of the particle. Then we can describe the evolution of the state trajectory of particle j in terms of the following *dynamic linear system*

$$X_j(k+1) = AX_j(k) + b_j(k), \tag{8}$$

where

$$\begin{aligned} A &= \begin{pmatrix} \chi w I & -\chi(cr + c_g r_g) I \\ \\ \chi w I & [1 - \chi(cr + c_g r_g)]\, I \end{pmatrix} \in \mathrm{I\!R}^{2n \times 2n}, \\ b_j(k) &= \begin{pmatrix} \chi(crp_j^k + c_g r_g p_g^k) \\ \\ \chi(crp_j^k + c_g r_g p_g^k) \end{pmatrix} \in \mathrm{I\!R}^{2n}. \end{aligned}$$

Since (8) is a linear and stationary system (being A constant), we can easily describe the trajectory $\{X_j(k)\}$ in the state space $\mathrm{I\!R}^{2n}$. Specifically, we can explicitly write (see, e.g., [1] and [11])

$$X_j(k) = X_{jL}(k) + X_{jF}(k), \qquad k \geq 0, \tag{9}$$

where $X_{jL}(k) \in \mathrm{I\!R}^{2n}$ is the so-called *free response* and $X_{jF}(k) \in \mathrm{I\!R}^{2n}$ the *forced response.* The free response and the forced response are respectively given by

$$X_{jL}(k) = A^k X_j(0), \qquad X_{jF}(k) = \sum_{\tau=0}^{k-1} A^{k-\tau-1} b_j(\tau). \tag{10}$$

Then, we can immediately observe that $X_{jL}(k)$ uniquely depends on the initial point $X_j(0)$ and it is not directly affected by the vectors p_j^k and p_g^k. On the other hand, $X_{jF}(k)$ does not depend directly on $X_j(0)$, while it relies on the p_j^k and p_g^k (see the above expression of $b_j(k)$). Since $X_{jL}(k)$ is so strongly dependent on $X_j(0)$, we can think about a suitable initialization of the j-th particle, such that the free response $X_{jL}(k)$ has specific properties.

Specifically, in the next section we will show that we can set $X_j(0)$ so that the components of the free response are in some sense *orthogonal.* This fact will hopefully make the overall trajectories of particles be better *scattered* in the search space (even if not necessarily orthogonal), and hence the PSO is expected to better detect some optima in the search space, without altering PSO iteration. We will support this heuristic argument with some numerical evidence.

Here, we stress that our proposal does not contribute to make PSO iteration globally convergent to stationary points. Nevertheless, it may improve the PSO performance, by means of possibly

(i) accelerating PSO progress towards an appropriate global minimizer, with respect to a standard PSO initialization;

(ii) provide a better approximate minimizer within the early iterations.

In the remaining part of this section, we investigate the structural properties of matrix A^k, so that we explicitly calculate its eigenpairs (see also [1]) that we will use for computing the free response $X_{jL}(k)$.

Proposition 1 *Suppose Assumptions 1 hold, and consider conditions (6g) and (6h). Then, the eigenvalues of matrix A in (8) have a modulus smaller than one.*

An immediate consequence of the above proposition is that we have

$$\lim_{k \to \infty} X_{jL}(k) = 0. \tag{11}$$

In other words, for a proper choice of the PSO coefficients the free response will always be eventually negligibly small. As a consequence, the effects of the free response are appreciable only for moderate values of k, which agrees with the overall spirit of heuristic procedures, that are expected to be effective in their early iterations. Differently, if the PSO coefficients satisfy Assumptions 1 except conditions (6g) and (6h), then the PSO might either diverge or converge to points that, in general, are not stationary and depend on $X_j(0)$.

Given the Assumptions 1, it is trivial to verify the two following facts. Matrix A has at most two distinct eigenvalues λ_1 and λ_2, each with algebraic multiplicity equal to n. The eigenvalues λ_1 and λ_2 are distinct if and only if assumption (6h) holds. In addition, some cumbersome but still trivial algebraic passages allow to express the free response $X_{jL}(k)$ as

$$X_{jL}(k) = A^k X_j(0) = \left[\begin{array}{c} \gamma_1(k)\nu_j^0 - \gamma_2(k)\xi_j^0 \\ \\ \gamma_3(k)\nu_j^0 - \gamma_4(k)\xi_j^0 \end{array} \right], \tag{12}$$

where, the coefficients $\gamma_1(k)$, $\gamma_2(k)$, $\gamma_3(k)$ and $\gamma_4(k)$ are

$$\begin{cases} \gamma_1(k) = \dfrac{\lambda_1^k(\chi w - \lambda_2) - \lambda_2^k(\chi w - \lambda_1)}{\lambda_1 - \lambda_2}, \\ \\ \gamma_2(k) = \dfrac{\chi(cr + c_g r_g)(\lambda_1^k - \lambda_2^k)}{\lambda_1 - \lambda_2}, \\ \\ \gamma_3(k) = \dfrac{(\lambda_1^k - \lambda_2^k)(\chi w - \lambda_1)(\chi w - \lambda_2)}{\chi(cr + c_g r_g)(\lambda_1 - \lambda_2)}, \\ \\ \gamma_4(k) = \dfrac{\lambda_1^k(\chi w - \lambda_1) - \lambda_2^k(\chi w - \lambda_2)}{\lambda_1 - \lambda_2}. \end{cases} \tag{13}$$

The above expressions are well-posed since by (6i) we have $\lambda_1 \neq \lambda_2$.

4 A novel initialization for particles, in an extended search space

In this section we exploit the characteristics of the free response $X_{jL}(k)$ in (10), to propose a novel initialization for PSO particles $j = 1, \ldots, P$. We describe how to define the vectors $X_j(0)$ in such a way that the exploration of the search space can be possibly improved.

Our approach is motivated by both theoretical and numerical results in the literature concerning derivative-free optimization methods, like the Pattern Search and Line Search-based ones in [12]. These algorithms iteratively explore the possible solutions on linear manifolds generated by the span of sets of search directions. The typical sets adopted are the so-called Positively Spanning Sets (PSS).

Definition 1 ([12]) *Let $d_i \in \mathbb{R}^n$, $i = 1, \ldots, p$, $p \geq n+1$. A set $D = \{d_1, \ldots, d_p\}$ is a Positively Spanning Set, if for any vector $y \in \mathbb{R}^n$ there exist nonnegative coefficients $a_i(y) \geq 0$, $i = 1, \ldots, p$, such that*

$$y = \sum_{i=1}^{p} a_i(y) d_i, \qquad \forall y \in \mathbb{R}^n. \tag{14}$$

Usually any $y \in \mathbb{R}^n$ is represented as a linear combination of n vectors of a basis. Differently, a PSS, exploiting a larger number of vectors (at least $n + 1$), can express any $y \in \mathbb{R}^n$ as conic combination of its vectors. One of the most used and effective PSS in $\mathbb{R}^n$, considered by derivative-free optimization methods, is the *compass search set*

$$D_\oplus = \{e_1, -e_1, e_2, -e_2, \cdots, e_n, -e_n\},$$

where $p = 2n$, and e_i is the i-th unit vector. These methods explore the values assumed by the function along $D_{\oplus}$, which is made of $n+n$ orthogonal directions. We refer the interested reader to [12] for a deeper introduction on the importance of PSSs in the literature of exact derivative-free optimization.

In case of PSO, at iteration k, the velocity ν_j^k of particle j may be regarded as a search direction of this particle from its current position ξ_j^k, see (7). Then, the search directions of PSO, namely the velocities of the different particles, are in general not orthogonal, and hardly modifiable without strongly spoiling PSO iteration. Thus, our idea is that of assigning the initial positions/velocities of particles, so that we can possibly force the orthogonality in wide sense of the free responses $\{X_{jL}(k)\}$ in (9). This latter fact is expected to favorably help the exploration of the search space, by scattering of the entire particles' trajectories.

Since $\lim_{k\to\infty} X_{jL}(k) = 0$, our idea is expected to be more effective for small values of the step index k. This implicitly suggests that our proposal is mainly tailored for expensive applications, where the objective function evaluation is rather costly, time resources are scarce and a reduced number of function evaluations (or equivalently iterations) can be allowed.

In the following we first suggest the basic elements of our contribution and then we formalize them. From (12), we know that the initial position and the free response of particle j are respectively given by

$$X_j(0) = \begin{pmatrix} \nu_j^0 \\ \\ \xi_j^0 \end{pmatrix}, \qquad \text{and} \qquad X_{jL}(k) = A^k X_j(0) = \begin{pmatrix} \gamma_1(k)\nu_j^0 - \gamma_2(k)\xi_j^0 \\ \\ \gamma_3(k)\nu_j^0 - \gamma_4(k)\xi_j^0 \end{pmatrix}. \tag{15}$$

Keeping in mind (9), we can consider the first n entries of $X(k)_{jL}$, that is $\gamma_1(k)\nu_j^0 - \gamma_2(k)\xi_j^0$, as an "approximation" of the overall velocity ν_j^k of the particle j, especially for moderate values of k. Hereinafter, we refer to $\gamma_1(k)\nu_j^0 - \gamma_2(k)\xi_j^0$ as *free response velocity.* Hence, we can force the PSO to search directions that are approximately orthogonal for moderate values of k, by choosing the initial state values of the particles so that

$$\left[\gamma_1(k)\nu_j^0 - \gamma_2(k)\xi_j^0\right]^T \left[\gamma_1(k)\nu_h^0 - \gamma_2(k)\xi_h^0\right] = 0, \qquad \forall j, h \ \text{ s.t. } \ 1 \le j \ne h \le P. \tag{16}$$

The next theorem clarifies our aim more accurately, giving a precise indication on PSO initialization. We remark that in the next result we assume $P = 2n$, though without loss of generality, with trivial modifications the same result holds allowing $P \le 2n$.

Theorem 1 *Consider PSO iteration (7), let Assumptions 1 and Proposition 1 hold, and suppose $P = 2n$. Then, the free response velocities of the particles are orthogonal at iteration $k \ge 0$, if the following initialization is chosen for the i-th and the $(n+i)$-th particles, $i = 1, \dots, n$,*

$$\begin{pmatrix} \nu_i^0 \\ \\ \xi_i^0 \end{pmatrix} = \rho_i \begin{pmatrix} \gamma_2(k)e_i \\ \\ \gamma_1(k)e_i \end{pmatrix}, \qquad \rho_i \in \mathbb{R}\setminus\{0\}, \quad i = 1, \dots, n, \tag{17a}$$

$$\begin{pmatrix} \nu_{n+i}^0 \\ \\ \xi_{n+i}^0 \end{pmatrix} = \rho_{n+i} \begin{pmatrix} -\gamma_1(k)e_i \\ \\ \gamma_2(k)e_i \end{pmatrix}, \qquad \rho_{n+i} \in \mathbb{R}\setminus\{0\}, \quad i = 1, \dots, n. \tag{17b}$$

Proof

By simple substitution, relation (16) is equivalent to

$$\begin{aligned}
0 &= \left[(\gamma_1(k)I \quad -\gamma_2(k)I) \begin{pmatrix} \nu_j^0 \\ \\ \xi_j^0 \end{pmatrix} \right]^T \left[(\gamma_1(k)I \quad -\gamma_2(k)I) \begin{pmatrix} \nu_h^0 \\ \\ \xi_h^0 \end{pmatrix} \right] \\
&= \begin{pmatrix} \nu_j^0 \\ \\ \xi_j^0 \end{pmatrix}^T \left[\begin{pmatrix} \gamma_1(k)I \\ \\ -\gamma_2(k)I \end{pmatrix} (\gamma_1(k)I \quad -\gamma_2(k)I) \right] \begin{pmatrix} \nu_h^0 \\ \\ \xi_h^0 \end{pmatrix} \\
&= \begin{pmatrix} \nu_j^0 \\ \\ \xi_j^0 \end{pmatrix}^T \begin{bmatrix} (\gamma_1(k))^2 I & -\gamma_1(k)\gamma_2(k)I \\ \\ -\gamma_2(k)\gamma_1(k)I & (\gamma_2(k))^2 I \end{bmatrix} \begin{pmatrix} \nu_h^0 \\ \\ \xi_h^0 \end{pmatrix}. \qquad (18)
\end{aligned}$$

As the matrix

$$\Lambda = \begin{bmatrix} (\gamma_1(k))^2 I & -\gamma_1(k)\gamma_2(k)I \\ \\ -\gamma_2(k)\gamma_1(k)I & (\gamma_2(k))^2 I \end{bmatrix}$$

is symmetric, condition (18) indicates that (16) is fulfilled if and only if the vectors

$$\begin{pmatrix} \nu_j^0 \\ \\ \xi_j^0 \end{pmatrix}, \qquad \begin{pmatrix} \nu_h^0 \\ \\ \xi_h^0 \end{pmatrix} \qquad (19)$$

are *mutually conjugate* with respect to Λ (see, e.g., [13, 14, 15]), with $1 \leq j \neq h \leq P$. Hence, a sufficient (but not necessary) condition for relation (16) to be fulfilled, is that the vectors (19) coincide with two orthogonal eigenvectors of matrix Λ. Indeed, if z_a and z_b are orthogonal eigenvectors of Λ, respectively associated to the eigenvalues λ_a and λ_b, then we have

$$z_a^T \Lambda z_b = z_a^T(\lambda_b z_b) = \lambda_b z_a^T z_b = 0.$$

This implies that any $2n$ orthogonal eigenvectors of matrix Λ are also mutually conjugate with respect to Λ. As a consequence, to satisfy condition (18) it suffices to compute the eigenvectors of matrix Λ, and set the vectors in (19) as proportional to the latter eigenvectors. Simple algebraic passages indicate that

$$\mu_1 = 0, \qquad \mu_2 = (\gamma_1(k))^2 + (\gamma_2(k))^2$$

are the only two distinct eigenvalues of matrix Λ and

$$\begin{pmatrix} \gamma_2(k)e_i \\ \\ \gamma_1(k)e_i \end{pmatrix} \in \mathrm{I\!R}^{2n}, \qquad \begin{pmatrix} -\gamma_1(k)e_i \\ \\ \gamma_2(k)e_i \end{pmatrix} \in \mathrm{I\!R}^{2n}, \qquad i = 1, \ldots, n$$

are the corresponding $2n$ eigenvectors. ◊

We complete this section by remarking the following issues:

- in case $P \leq 2n$ the initialization (17) immediately applies. On the other hand, in case $P > 2n$, the first $2n$ particles can be initialized as in (17), while for the remaining $P - 2n$ particles any initialization from the literature can be set by the user, without altering our analysis;

- the conditions (16) hold for a given step $k \geq 0$ (which can be arbitrarily chosen by the user). In particular, in our numerical experience we set the initialization (17) at iteration $k = 0$. Since the free responses of the particles tend to asymptotically fade (being $\lim_{k\to\infty} X_{jL}(k) = 0$, for any particle

j), we expect that the orthogonality property guaranteed by Theorem 1, is more effective when it holds for a small value of k (say $k \leq 10$). This indeed guarantees that the contribution of the free responses of the particles is not negligibly small. The last consideration is even more evident on those problems where the entries of vectors $X_{jF}(k)$ in (9) tend to rapidly increase, with respect to those of $X_{jL}(k)$. In our test (portfolio) problems, that we introduce in the following section, the latter drawback is partially reduced, since the entries of vectors p_j^k and p_g^k tend to approach values in the range $[0, 1]$, which is the range of each asset investment;

- the choice of the nonzero coefficients ρ_i, ρ_{n+i}, in (17), is arbitrary, so that they can be set by the user on the basis of the problem in hand.

5 The complex portfolio selection problem

Generally speaking, making effective portfolio selection in real stock markets is not an easy task for at least the following two reasons:

- First, we need to gauge the risk by measures that both satisfy appropriate formal properties (namely, coherence) and better couple with the non-normal return distributions, which characterize the stock markets. Furthermore, it should be desirable that these risk measures were parameterized with respect to the investor's risk attitude;
- Second, we have to take into account several practises and rules of the portfolio management industry that can affect the portfolio selection process like, for instance, the use of bounds for the minimum and the maximum number of stocks to trade.

In order to deal with both such issues, in this paper we consider a recently proposed complex portfolio selection problem (see [2]). In this problem, as measure of risk of the portfolio return we consider a coherent risk measure based on the combination of lower and upper moments of different orders of the portfolio return distribution (see [16]). Such a measure shows to be able to manage non-Gaussian distributions of asset returns and to reflect different investors' risk attitudes. In particular, it permits to take into account both the risk contained in the "bad" tail (the left one of the portfolio return) and the chance contained in the "good" tail (the right one of the same portfolio return). Furthermore, as the professional practices and rules are concerned, the considered portfolio selection problem uses bounds for the minimum and the maximum number of stocks to trade (bounds known as *cardinality constraints*), and includes the minimum and the maximum capital percentage to invest in each asset. All these bounds are of interest for the fund managers in order to control, in an indirect way, the transaction costs.

In order to formalize the considered portfolio selection problem, we need to introduce the following notations:

- $x_i \in \mathbb{R}$, for $i = 1, \dots, N$, is the weight of the i-th asset in the portfolio, with $x^T = (x_1, \dots, x_N)$;
- $z_i \in \{0, 1\}$, for $i = 1, \dots, N$, is a variable such that $z_i = 1$ if the i-th asset is included in the portfolio, $z_i = 0$ otherwise, with $z^T = (z_1, \dots, z_N)$;
- $r = \sum_{i=1}^N r_i x_i$ is the random variable indicating the portfolio return, in which r_i, for $i = 1, \dots, N$, is the random variable indicating the return of the i-th asset;
- $\rho_{a,p}(r)$ is the coherent risk measure of the portfolio return, with $a \in [0, 1]$ and $p \in [1, +\infty)$ (note that appropriate values of a and p permit to suitably parameterize $\rho_{a,p}(r)$ with respect to the investor's risk attitude);

- $E[\cdot]$ indicates the expected value of the random variable argument;
- $y^- = \max\{0, -y\}$ and $y^+ = (-y)^-$;
- $\widehat{r}_i = E[r_i]$, $E[r] = \sum_{i=1}^N \widehat{r}_i x_i$, while r_e is the minimum desired expected return of the portfolio;
- K_d and K_u are the minimum and the maximum number of stocks to trade, respectively, with $1 \le K_d \le K_u \le N$;
- d and u are the minimum and the maximum capital percentage to invest in each asset, respectively.

Now, the overall complex portfolio selection problem can be formulated as follows:

$$
\begin{aligned}
\min_{x,z} \;\; & \rho_{a,p}(r) = a\|(r - E[r])^+\|_1 + (1-a)\|(r - E[r])^-\|_p - E[r] && \text{(20a)}\\
\text{s.t.} \;\; & E[r] \ge r_e && \text{(20b)}\\
& \sum_{i=1}^N x_i = 1 && \text{(20c)}\\
& K_d \le \sum_{i=1}^N z_i \le K_u && \text{(20d)}\\
& z_i d \le x_i \le z_i u, \qquad i = 1, \ldots, N, && \text{(20e)}\\
& z_i(z_i - 1) = 0, \qquad i = 1, \ldots, N. && \text{(20f)}
\end{aligned}
$$

In this formulation, (20b) is the constraint about the minimum desired expected return of the portfolio and (20c) is the usual budget constraint.

Note that the above formulation of the portfolio selection problem is nonlinear, nonconvex, nondifferentiable, mixed-integer and NP-hard [17].

6 Plan of our numerical experience

In this section we report the plan of our experimental analysis. We first consider the following auxiliary problem:

$$\min_{x \in \mathbb{R}^N, \; z \in \mathbb{R}^N} \quad P(x, z; \varepsilon), \tag{21}$$

where

$$
\begin{aligned}
P(x,z;\varepsilon) \;\; = \;\; & \rho_{a,p}(r) + \frac{1}{\varepsilon_0}\Bigg[\varepsilon_1 \max\left\{0, r_e - \sum_{i=1}^N \hat{r}_i x_i\right\} + \varepsilon_2 \left|\sum_{i=1}^N x_i - 1\right| + \varepsilon_3 \max\left\{0, K_d - \sum_{i=1}^N z_i\right\}\\
& + \varepsilon_4 \max\left\{0, \sum_{i=1}^N z_i - K_u\right\} + \varepsilon_5 \sum_{i=1}^N \max\left\{0, z_i d - x_i\right\}\\
& + \varepsilon_6 \sum_{i=1}^N \max\left\{0, x_i - z_i u\right\} + \varepsilon_7 \sum_{i=1}^N |z_i(1 - z_i)|\Bigg]
\end{aligned}
$$

and $\varepsilon = (\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_7)^T > 0$ is a set of penalty parameters. As discussed in Section 8, problem (21) can be seen as an unconstrained reformulation of problem (20), as the two problems present to large extent equivalent solutions (see Section 8), given an opportune choice of the penalty vector ε.

Then, we have tested our novel PSO initialization to determine the solutions of problem (20). Specifically, we used PSO to solve the auxiliary problem (21), and by setting the values of the penalty vector ε according to two different frameworks (see Section 6.1):

(1) A parameter control approach that uses an adaptive penalty mechanism, in which the set of penalty parameters is iteratively updated during the search (see also [18]);

(2) A parameter tuning approach (REVAC, see [19, 20]), in which penalty parameters values are set before the algorithm run.

Finally, to obtain a reference value of problem (20) for the above approaches, we also reformulated (20) as a nonlinear mixed-integer problem with $2N + 2T$ unknowns, that we solved through a standard exact solver. Since our portfolio selection problem is NP-hard, when the number of assets increases exact solvers may require a prohibitive amount of time for computation, which definitely may discourage their use by traders. Details of both the mixed-integer reformulation and the relative solver adopted are reported in Section 9.

As for the numerical instances, we considered assets belonging to stock exchange indexes, in which daily close prices over a time horizon T are converted in daily returns by using the formula $r_{i,t} = \log\left(\frac{S_{i,t}}{S_{i,t-1}}\right)$, where $S_{i,t}$ represents the price of asset i at time t and $r_{i,t}$ represents the return of asset i at time t. Then we approximate the expected values that appear in the objective function (20a) of our portfolio selection problem (20) with the associated sample means in accordance with [2, 16]:

$$\|(r - E[r])^+\|_1 = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i\right)^+$$

$$\|(r - E[r])^-\|_p = \frac{1}{T^{\frac{1}{p}}}\left(\sum_{t=1}^{T}\left[\left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i\right)^-\right]^p\right)^{\frac{1}{p}}.$$

Hence, we write:

$$\rho_{a,p}(r) = \frac{a}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i\right)^+ + \frac{1-a}{T^{\frac{1}{p}}}\left(\sum_{t=1}^{T}\left[\left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i\right)^-\right]^p\right)^{\frac{1}{p}} - \sum_{i=1}^{N}\hat{r}_i x_i. \tag{22}$$

6.1 Penalty vector settings

In this subsection we describe the two settings that we used to solve real instances of the portfolio problem (20). In this context, let us stress that the purpose of our numerical experience was that of testing the PSO initialization (17). Thus, here we are

- neither interested to propose a refinement of either the two approaches above,
- nor we want to compare their performance in order to propose a preferable strategy.

On the contrary, we want to verify whether the proposal (17) is a winning PSO initialization for both approaches, with respect to the standard one used in the literature. The results of the experiments, described in Section 7, where PSO was implemented in MATLAB 7, indicate that our proposal is robust with respect to the different strategies adopted.

Exact Penalty approach with dynamic parameters update. This setting determines the solution of problem (20) by adaptively updating the value of the vector of penalty parameters ε, when using PSO for the solution of the auxiliary problem (21) (motivations and advantages for adaptively updating the parameters are discussed in Section 8). In this framework ε is updated at each iteration k of PSO. Hence, hereinafter, we use of the symbol ε^k to indicate the value of vector ε at iteration k.

For $k = 0$ the initial parameters vector ε^0 is set as

$$\varepsilon^0 = \begin{pmatrix} 10^{-4} \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \in \mathbb{R}^8,$$

i.e. $\varepsilon_0^0 = 10^{-4}$ and $\varepsilon_i^0 = 1$, $i = 1, \ldots, 7$. The latter choice guarantees an equal penalization (i.e. $\varepsilon_2^0 = \cdots = \varepsilon_7^0$) for all constraints violations, while the value of ε_0^1 is chosen much smaller than all the constraints violations. Observe that from (4) p_g^k is the best position of PSO particle at iteration k, when solving (21). Thus, recalling that the problem (21) has $2N$ unknowns (namely the vectors x and z), we use the notation

$$\begin{pmatrix} x_g^k \\ \\ z_g^k \end{pmatrix} = p_g^k$$

to define the subvectors x_g^k and z_g^k.

On the other hand, for $k \geq 1$, the vector ε^k is updated according to a twofold strategy. We update ε_0^k by checking for a possible improvement (i.e. a decrease) of the value of $\rho_{a,p}(r_g^k)$, where $r_g^k = \sum_{i=1}^N \left(x_g^k\right)_i r_i$. We update ε_i^k, $i = 1, \ldots, 7$, by checking for the violation of the constraints:

$$v_1(x_g^k, z_g^k) = \max\left\{0, r_e - \sum_{i=1}^N \hat{r}_i \left(x_g^k\right)_i\right\}$$

$$v_2(x_g^k, z_g^k) = \left|\mathbf{1}^T x_g^k - 1\right|$$

$$v_3(x_g^k, z_g^k) = \max\{0, K_d - \mathbf{1}^T z_g^k\}$$

$$v_4(x_g^k, z_g^k) = \max\{0, \mathbf{1}^T z_g^k - K_u\}$$

$$v_5(x_g^k, z_g^k) = \sum_{i=1}^N \max\left\{0, \left(z_g^k\right)_i d - \left(x_g^k\right)_i\right\}$$

$$v_6(x_g^k, z_g^k) = \sum_{i=1}^N \max\left\{0, \left(x_g^k\right)_i - \left(z_g^k\right)_i u\right\}$$

$$v_7(x_g^k, z_g^k) = \sum_{i=1}^N \left|\left(z_g^k\right)_i \left(1 - \left(z_g^k\right)_i\right)\right|,$$

with $\mathbf{1} = (1, \cdots, 1)^T \in \mathbb{R}^N$. More in particular, as a general rule we adopted the following practical strategy (the choice of the coefficients is motivated by efficiency reasons, over our portfolio selection problems):

- every 20 iterations of PSO we possibly update the entry ε_0^{k+1} of ε^{k+1}, according with the following rule:

$$\varepsilon_0^{k+1} = \begin{cases} \min\left\{3\varepsilon_0^k, 1\right\} & \text{if } \rho_{a,p}(r_g^k) \geq \rho_{a,p}(r_g^{k-1}) \\ \max\left\{0.6\varepsilon_0^k, 10^{-15}\right\} & \text{if } \rho_{a,p}(r_g^k) < 0.90 \cdot \rho_{a,p}(r_g^{k-1}) \\ \varepsilon_0^k & \text{otherwise;} \end{cases} \tag{23}$$

- every 40 iterations of PSO we update the entries ε_i^{k+1}, $i = 1, \ldots, 7$, of ε^{k+1}, according with the following rule:

$$\varepsilon_i^{k+1} = \begin{cases} \min\left\{2\varepsilon_i^k, 10^4\right\} & \text{if } v_i(x_g^k, z_g^k) > 0.95 \cdot v_i(x_g^{k-1}, z_g^{k-1}) \\ \max\left\{\frac{1}{2}\varepsilon_i^k, 10^{-4}\right\} & \text{if } v_i(x_g^k, z_g^k) < 0.90 \cdot v_i(x_g^{k-1}, z_g^{k-1}) \\ \varepsilon_i^k & \text{otherwise.} \end{cases} \tag{24}$$

Roughly speaking, by relations (23) if the risk functional $\rho_{a,p}(r_g^k)$ increases, then in (21) a smaller penalty coefficient is imposed on constraints violation, i.e. ε_0^{k+1} must increase. With a similar reasoning, when $\rho_{a,p}(r_g^k)$ decreases then ε_0^{k+1} is allowed to decrease, in order to improve feasibility. As regards (24), if the i-th violation $v_i(x_g^k, z_g^k)$ significantly increases with respect to $v_i(x_g^{k-1}, z_g^{k-1})$ (i.e. we are possibly pursuing optimality while worsening feasibility), then the corresponding penalty parameter ε_i^{k+1} is increased accordingly. Conversely, with an opposite rationale, in case we observe a relevant improvement of feasibility (i.e. $v_i(x_g^k, z_g^k) \ll v_i(x_g^{k-1}, z_g^{k-1})$), then the parameter ε_i^{k+1} is decreased.

Approach with Parameter Tuning by REVAC. This setting approaches the solution of problem (21) by tuning a-priori the value of the vector of penalty parameters ε, before solving (21). In this framework, we ran a set of preliminary experiments using REVAC. These experiments allowed us to tune the parameters $w_1, \ldots, w_7$, being $w_i = \varepsilon_i/\varepsilon_0$ and $\varepsilon = (\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_7)^T$ is the penalty vector in (21). REVAC [19, 20] is an Estimation of Distribution Algorithm used to tune the parameters of an algorithm. It relies on information theory to measure parameter relevance. Roughly speaking, for each parameter set REVAC considers a value distribution that assigns high probabilities to values leading to a good compromise between the algorithm performance and the algorithm complexity. REVAC uses as a measure of complexity the Shannon entropy.

REVAC is an iterative algorithm. Initially, it creates a uniform distribution over the parameters space that it iteratively refines (i.e. smooths). This is done by an evolutionary process that starts from an initial parameter vector population. Then, it generates new parameter vectors by choosing the best subset of vectors with respect to expected performance, in order to replace the eldest individuals in the population [21]. The smoothing feature is assured by an operator that defines a mutation interval for each parameter. At each iteration, it sorts the current population parameter values and defines a new distribution by deleting a given number of extreme values. Then, it uses this new distribution to draw the next population parameter values randomly. The Shannon entropy is supposed to decrease over the iterations [3], and we can use the information gathered to infer information on the parameters. Namely, parameters that show a great decrease of entropy are likely the most sensitive to their values, hence they are the most promising for parameter value choices, and are the parameters whose distribution likely shows the presence of narrow peaks ([3]).

In our setting, we first ran REVAC to understand the relative relevance of parameters, and we identified the 2 most sensitive parameters by selecting the 2 parameters whose entropy most decreases over the REVAC execution. Hence, we used the outcome of this run to set the values of the 5 remaining parameters, by using the median of their resulting distributions. Then, we re-ran REVAC (using these latter parameters values) and at the end of the run we defined the values of the 2 most sensitive parameters, by setting them to the median of their resulting distributions.

As regards the code of REVAC we used, the MATLAB implementation by Volker Nannen was adopted, which is publicly available at `http://www.complexity-research.org/revac`.

6.2 Our benchmarks for portfolio problems

Considering the numerical experience we have performed over portfolio selection problems, in this paper we preferred to report only the results relative to a reduced (but representative) test set. Indeed, we report experiments over two instances coming from real stock-exchange data: the first is drawn from the FTSE MIB Index (daily data from January 2009 to July 2014), the latter is drawn from the Nikkei Index (daily data from January 2008 to May 2014). Benchmark features are summarized in Table 1, together with the PSO standard parameters used to minimize the unconstrained reformulation (21) of problem (20), for each instance.

We remark that the two benchmarks were chosen in order to be representative of opposite real situations. In particular, complete information is available for the first one, and just a few daily data are missing. Differently, the set of data is incomplete for the second benchmark, as a large number of daily values are unknown. Here our aim is to compare our novel initialization versus the standard PSO one, considering these two opposite situations. We will show that an appreciable improvement in the early iterations of PSO is obtained using (17), in both cases.

Table 1: Benchmarks features and PSO parameters used for the instances in Section 6.2.

Benchmarks parameters		**FTSE MIB**	**Nikkei**
Name	**Meaning**		
T	Time horizon	1396	1664
N	Number of assets	32	225
r_e	Minimum desired expected return	0.0005	0
K_d	Maximum number of assets in portfolio	4	4
K_u	Minimum number of assets in portfolio	28	28
d	Smallest fraction of each asset in portfolio	0.05	0.05
u	Largest fraction of each asset in portfolio	1	1
a	Parameter of the risk functional	0.5	0.5
p	Parameter of the risk functional	3	3
PSO parameters			
c_j^k		1.49618	1.49618
c_g^k		1.49618	1.49618
w^k		0.7298	0.7298
χ^k		1	1

For each instance we have set a minimum required return r_e in (20b), over the observed period T associated with the benchmark in hand. As for the cardinality constraint, we have imposed a lower and upper bound (namely K_d and K_u in (20d)), according with the current literature of metaheuristics for portfolio selection (see [22]), which are reasonable for practical scenarios.

7 Numerical results

Here we report the main results and statistics obtained by running the two approaches detailed in Section 6, comparing them with a PSO scheme that implements the *standard* (i.e., random) initialization in the literature. For the sake of completeness, we also report in this section additional numerical results obtained from the nonlinear mixed-integer model described in Section 9.

Exact Penalty approach with dynamic parameters update. Figures 1 and 2 report respectively the values of $P(x_g^k, z_g^k; \varepsilon^k)$ and $\rho_{a,p}(r_g^k)$ (see also Section 6.1) in (21), obtained for a typical PSO run when FTSE MIB data are considered. In Figures 1 and 2 the quantities are obtained in the first 150 iterations of PSO, respectively using (17) vs. a standard PSO initialization. These results confirm that using our proposal we obtain a better minimization in the early PSO iterations (say $k \leq 150$). For the sake of completeness, on this instance after 2000 PSO iterations we obtain (using the novel initialization) the final values

return:	0.00088
$P(x_g^{2000}, z_g^{2000}; \varepsilon^{2000})$:	32.991
$\rho_{a,p}(r_g^{2000})$:	0.015968

The latter figures show also how our penalty approach is effective in both minimizing (20a) and improving the feasibility of the current best PSO solution (x_g^k, z_g^k). We argue that the constant progress of our proposal, in Figures 1 and 2, is mainly due to the joint use of our initialization (17) and a proper penalty parameters update. On this guideline, observe that the parameters update in (23)-(24) is performed only at intervals of 20 and 40 PSO iterations, respectively, and in Figure 1 the *tapered* shape of plots corresponds to a change of the penalty parameters. In particular, the plot with our new initialization has evident updates every 20 iterations (when possibly only ε_0^k is updated), and even drastic updates every 40 iterations (when possibly the entire vector ε^k is updated).

As regards the Nikkei instance, we recall that as indicated in Section 6.2 poor information is available as several data are missing. The poor quality of input data is expected to affect the final results. Indeed, no reliable solution is obtained with this approach, in a reasonable computational time. Nevertheless, Figures 3-4 seem to confirm that even in this situation, in the early PSO iterations our novel initialization of PSO provides better results with respect to the standard one in the literature.

Approach with Parameter Tuning by REVAC. First note that the REVAC approach is inherently stochastic. The weights w_i, $i = 1, \ldots, 7$ (see again Section 6.1) are assessed using statistical techniques, therefore average results should be reported, after performing a number of different runs. However, note that similar results are obtained from one run to another. The latter fact justifies our choice to report, without loss of generality, only the results relative to one run. On this purpose, likewise the Exact Penalty approach with dynamic parameters update, Figures 5 and 6 report the values of $P(x_g^k, z_g^k; \varepsilon)$ and $\rho_{a,p}(r_g^k)$, respectively, obtained in the early 150 iterations of a typical PSO run, using REVAC, when FTSE MIB data are considered. Again, these results give evidence that our proposal represents a winning strategy in the early PSO iterations (say $k \leq 150$).

As regards the Nikkei instance using REVAC, we obtain the results in Figures 7 and 8, which yield analogous conclusions with respect to FTSE MIB data, even though the Nikkei instance contains poor information.

For the sake of completeness, we also report directly in Tables 2 and 3 the main statistics about the fitness $P(x_g^k, z_g^k; \varepsilon)$ and the risk functional $\rho_{a,p}(r_g^k)$, along with the success ratio (i.e., the percentage of all-feasible solutions found) and the mean computation time over 30 runs. We recall that we do not claim our proposal to be a winning strategy for large values of the number of iterations. Thus, *a fortiori* we did not expect our new PSO initialization to provide better asymptotic results on the poor Nikkei instance. Nevertheless, the data in Tables 2-3 show that our novel initialization asymptotically compares fairly well with the basic PSO results.

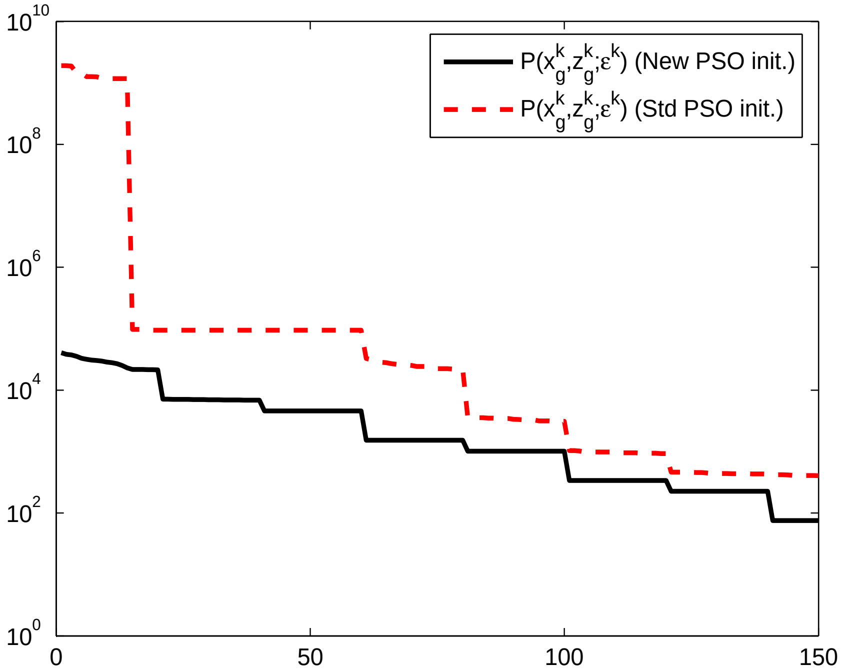

Figure 1: The values of $P(x_g^k, z_g^k; \varepsilon^k)$ in (21), for the first 150 PSO iterations on the *FTSE MIB* instance, using the Dynamic Parameters Update. A logarithmic scale is used for the ordinate axis.

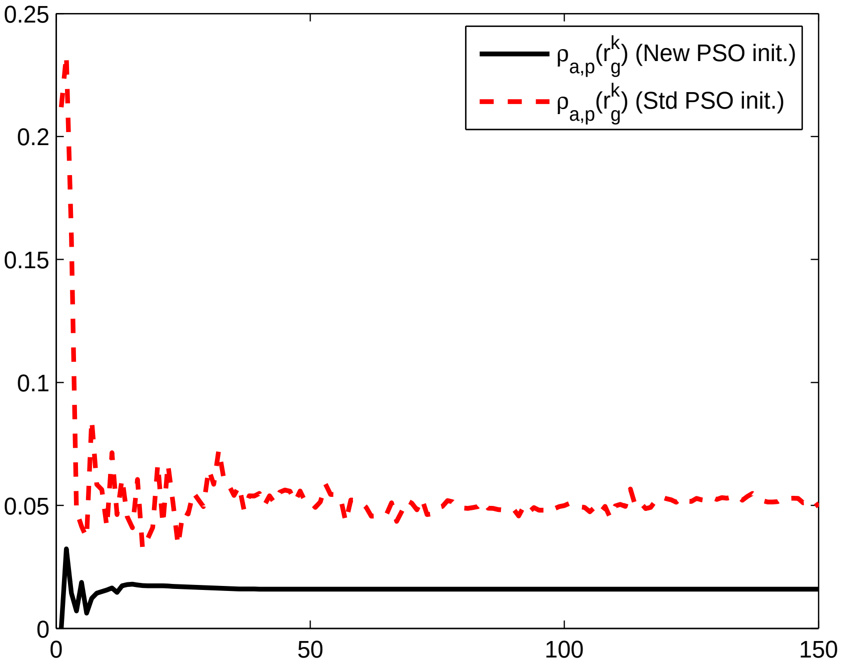

Figure 2: The values of $\rho_{a,p}(r_g^k)$ in (21), for the first 150 PSO iterations on the *FTSE MIB* instance, using the Dynamic Parameters Update.

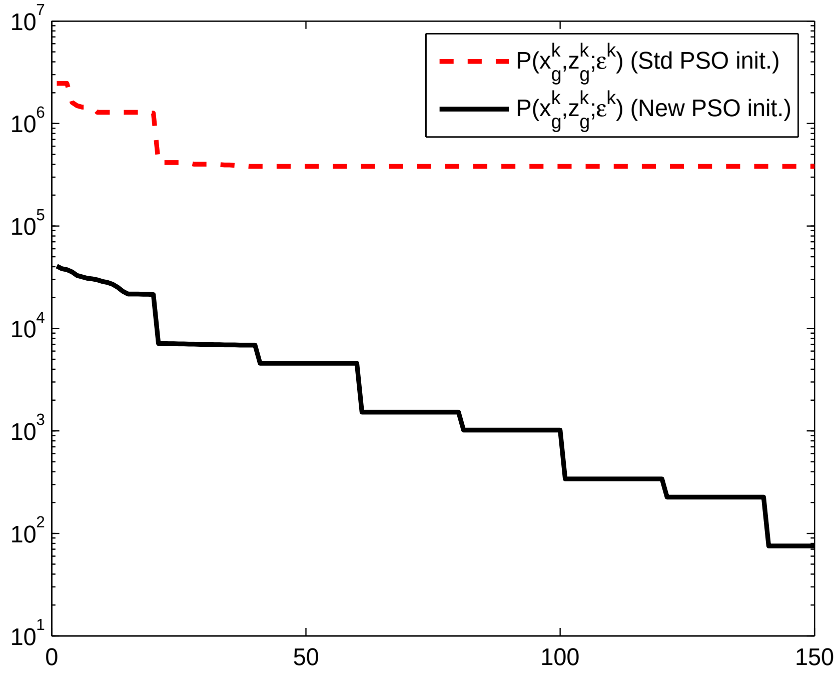

Figure 3: The values of $P(x_g^k, z_g^k; \varepsilon^k)$ in (21), for the first 150 PSO iterations on the *Nikkei* instance, using the Dynamic Parameters Update. A logarithmic scale is used for the ordinate axis.

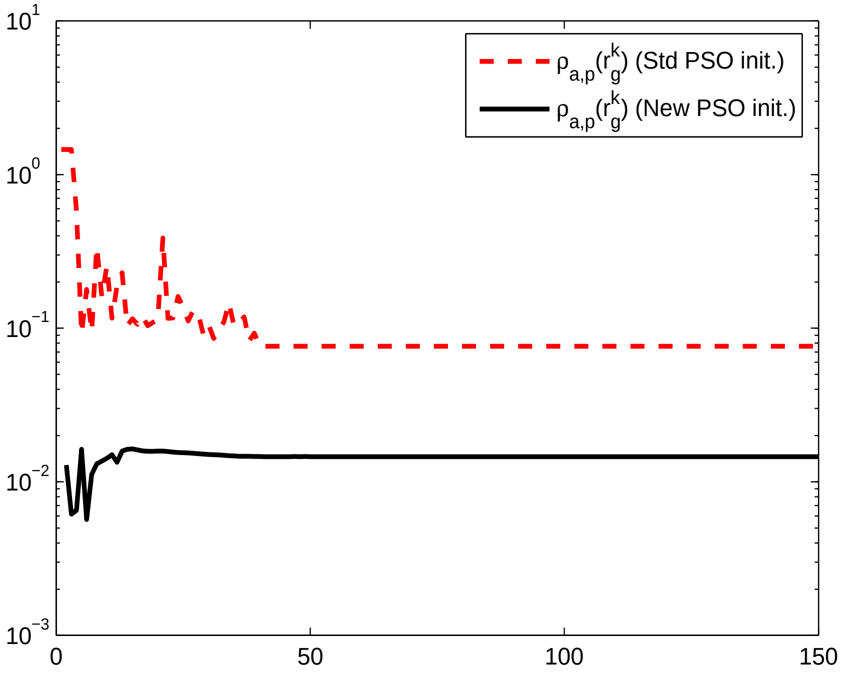

Figure 4: The values of $\rho_{a,p}(r_g^k)$ in (21), for the first 150 PSO iterations on the *Nikkei* instance, using the Dynamic Parameters Update.

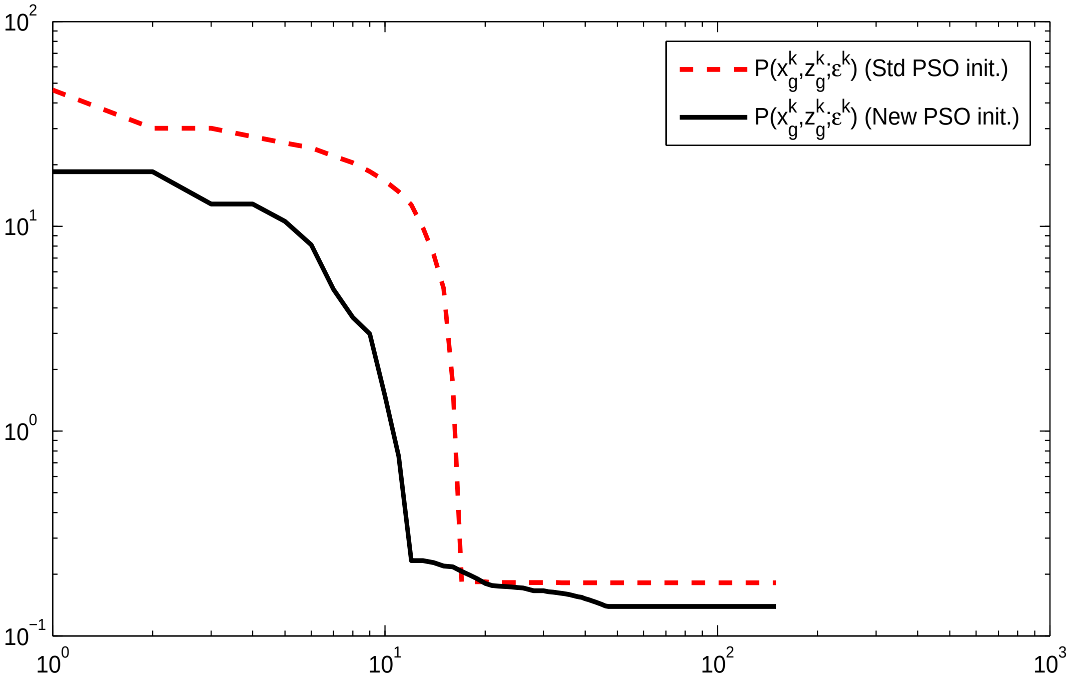

Figure 5: The values of $P(x_g^k, z_g^k; \varepsilon^k)$ in (21), for the first 150 PSO iterations on the *FTSE MIB* instance, using REVAC. A logarithmic scale is used for the both the abscissa and the ordinate axis.

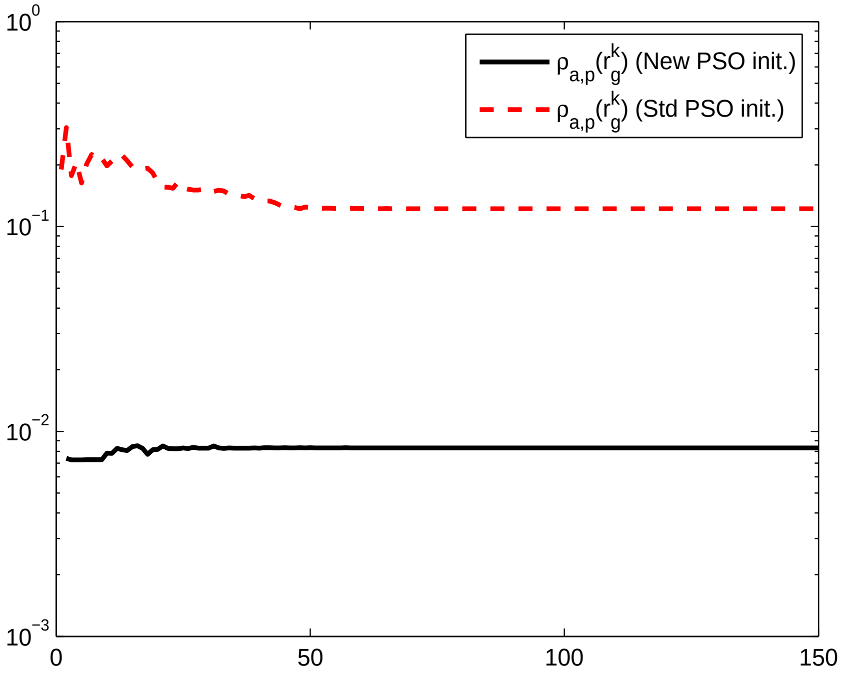

Figure 6: The values of $\rho_{a,p}(r_g^k)$ in (21), for the first 150 PSO iterations on the *FTSE MIB* instance, using REVAC. A logarithmic scale is used for the ordinate axis.

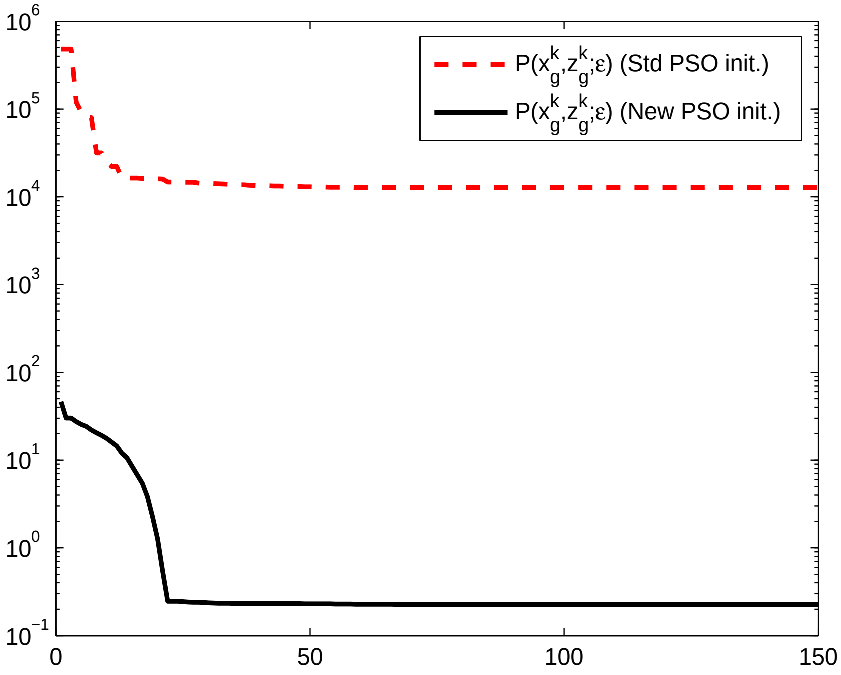

Figure 7: The values of $P(x_g^k, z_g^k; \varepsilon^k)$ in (21), for the first 150 PSO iterations on the *Nikkei* instance, using REVAC. A logarithmic scale is used for the ordinate axis.

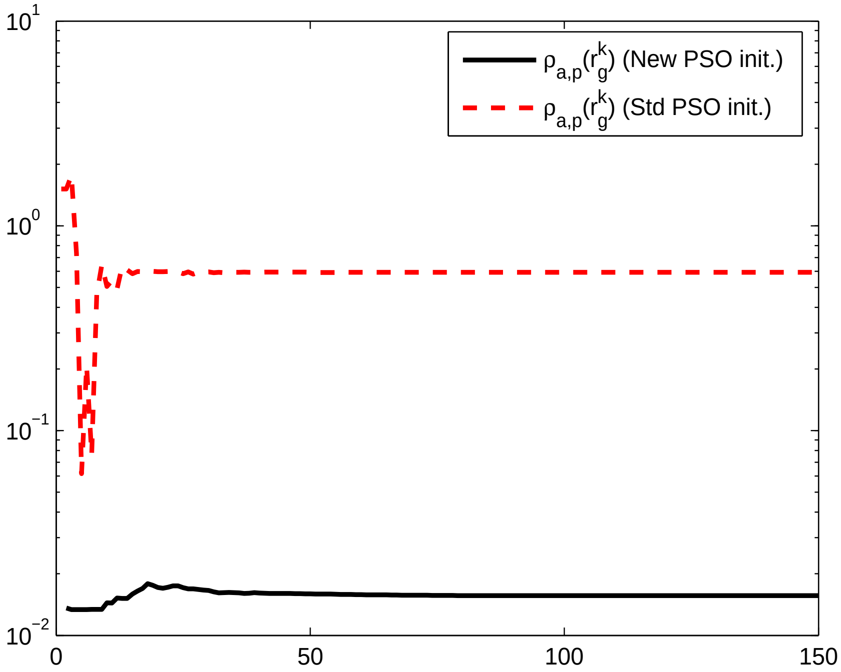

Figure 8: The values of $\rho_{a,p}(r_g^k)$ in (21), for the first 150 PSO iterations on the *Nikkei* instance, using REVAC. A logarithmic scale is used for the ordinate axis.

Table 2: Experimental Results, FTSI MIB instance, REVAC settings.

	Fitness			$\rho_{a,p}$			**Time**	**Success ratio**
Strategy	**Min**	**Max**	**Std dev**	**Min**	**Max**	**Std dev**	**Mean**	**Mean**
PSO-newinit-REVAC	0.011803	0.011803	0	0.008915	0.008915	0	463.05	0
PSO-standard	0.007027	0.233835	0.067829	0.006912	0.155254	0.07204026	458.47	23.3
NEOS (FilMINT)	-	-	-	0.006582	0.006582	-	506.16	100

Table 3: Experimental Results, NIKKEI instance, REVAC settings.

	Fitness			$\rho_{a,p}$			**Time**	**Success ratio**
Strategy	**Min**	**Max**	**Std dev**	**Min**	**Max**	**Std dev**	**Mean**	**Mean**
PSO-newinit-REVAC	0.2254	0.2254	0	0.0156	0.0156	0	831.05	0
PSO-standard	0.0367	0.7629	0.2350	0.3446	0.9272	0.2674	818.72	0
NEOS (FilMINT)	-	-	-	0.002299	0.002299	-	7064.64	100

8 Some insights on the nonsmooth reformulation

Since we are interested to apply an exact penalty approach to solve the portfolio problem (20), similarly to [2], here we preliminarily:

- detail some basics on the class of exact penalty functions we are going to use;
- motivate our proposal for a nonsmooth reformulation of (20) by a proper exact penalty function.

We are persuaded about the importance of this section, in order to provide a more complete technical support to the reader, and for a better comprehension of the iterative parameters update (23)-(24). However, those who are mainly interested about the application of our proposal, might possibly skip some of the contents of this section, and directly focus on the nonsmooth reformulation (21) of problem (20).

For a correct analysis, in the sequel we prefer to introduce a specific taxonomy, in order to avoid possible misunderstandings, since in the literature different types of exactness can be set for a given penalty function, depending on the constraint qualification condition adopted. In particular, to simplify our results, but preserving more generality with respect to [2], we give the next definitions.

Definition 2 (KKT point) *Let $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$, $g : \mathbb{R}^n \to \mathbb{R}^p$, with $f, h, g \in \mathcal{C}^1(\mathbb{R}^n)$, and consider the constrained problem*

$$\min_{y \in \mathcal{F}} \; f(y), \tag{25}$$

where

$$\mathcal{F} = \{y \in \mathbb{R}^n \; : \; h(y) = 0, \; g(y) \leq 0\}.$$

Let $\mathcal{F}$ be compact. We say that the point $(\bar{y}, \bar{\mu}, \bar{\lambda}) \in \mathbb{R}^{n+m+p}$, with $\bar{y} \in \mathcal{F}$, is a Karush-Kuhn-Tucker (KKT) point for problem (25) if it satisfies the following first order stationarity conditions

$$\nabla f(\bar{y}) + \bar{\mu}^T \nabla g(\bar{y}) + \bar{\lambda}^T \nabla h(\bar{y}) = 0$$

$$\bar{\mu}^T g(\bar{y}) = 0$$

$$\bar{\mu} \geq 0.$$

Definition 3 (MFCQ) *Given the problem (25), we say that at point $\hat{y} \in \mathbb{R}^n$ the Mangasarian-Fromowitz Constraint Qualification (MFCQ) condition holds if* *(a) the vectors $\nabla h_1(\hat{y}), \ldots, \nabla h_m(\hat{y})$ are linearly independent;*

(b) there exists a nonzero vector $d \in \mathbb{R}^n$ such that

1. $\nabla h_i(\hat{y})^T d = 0$, for $i = 1, \ldots, m$

2. $\nabla g_j(\hat{y})^T d < 0$, for $j = 1, \ldots, p$.

For a more complete analysis we could introduce other constraint qualification conditions as the well-known Linear Independence Constraint Qualification (LICQ), which is among the most used in the literature (see also [2]). However, since MFCQ is *weaker* than LICQ, by assuming MFCQ we will gain more generality while maintaining a simple analysis.

Now, we can introduce the parameter dependent problem

$$\min_{y \in \overset{\circ}{\Omega}} \; P(y;\varepsilon) \tag{26}$$

where $P(y;\varepsilon)$ is a penalty function which depends on $f(y)$ and $\mathcal{F}$, $\varepsilon \in \mathbb{R}$ with $\varepsilon > 0$, and we assume that $\Omega = cl\left(\overset{\circ}{\Omega}\right)$, where $\overset{\circ}{\Omega}$ contains the compact set $\mathcal{F}$ in (25), i.e.

$$\overset{\circ}{\Omega} \supset \mathcal{F}. \tag{27}$$

We are interested about inferring information on the global solutions of (25), by means of solving (26). Since in general the solutions (either local or global) of (26) are obtained by means of iterative descent methods, starting from the initial guess $\bar{y}$, the choice of Ω is essentially equivalent to the choice of $\bar{y}$ (see also (28)). Observe that any exact penalty function approach has precisely the expression in (26), for a suitable choice of the function $P(y;\varepsilon)$, the parameter ε and the set Ω. Moreover, since $\overset{\circ}{\Omega}$ does not contain any point of its boundary, a local solution (if any) of (26) is an unconstrained minimizer of $P(y;\varepsilon)$. Thus, establishing a complete relation between the solutions of (25) and (26) might be of great interest, inasmuch as the constrained problem (25) can be solved by efficient algorithms for the unconstrained problem (26). This spots light on the fact that iteratively solving the problem

$$\min_{y \in \mathbb{R}^n} \; P(y;\varepsilon),$$

starting from the initial point $\bar{y}$ and for a given choice of the real parameter $\bar{\varepsilon} > 0$, *does not in general yield a solution of* (25), because the level set

$$\mathcal{L}(P,\bar{y},\bar{\varepsilon}) = \{y \in \mathbb{R}^n \;:\; P(y;\bar{\varepsilon}) < P(\bar{y};\bar{\varepsilon})\} \tag{28}$$

possibly does not satisfy the condition $\mathcal{L}(P,\bar{y},\bar{\varepsilon}) \supseteq \mathcal{F}$, i.e. condition (27). This is one of the main reasons for which exact penalty approaches have always been, since their early versions, so strongly dependent on the choice of the penalty parameter, even on convex problems. As a simple numerical example of the latter drawback, consider the convex quadratic problem

$$\begin{aligned} \min \; & y^2 \\ & y \geq 3, \end{aligned} \tag{29}$$

whose solution is clearly $y^* = 3$. Consider also the associated nondifferentiable exact penalty problem

$$\min_{y \in \Omega} \; y^2 + \frac{1}{\varepsilon}\max\{0, 3-y\} \tag{30}$$

Here the penalty function we choose is $P(y;\varepsilon) = y^2 + 1/\varepsilon \max\{0, 3-y\}$, and is clearly nondifferentiable in $y = 3$, with the shape in Figure 9, for two different values of the parameter ε. Observe that even setting

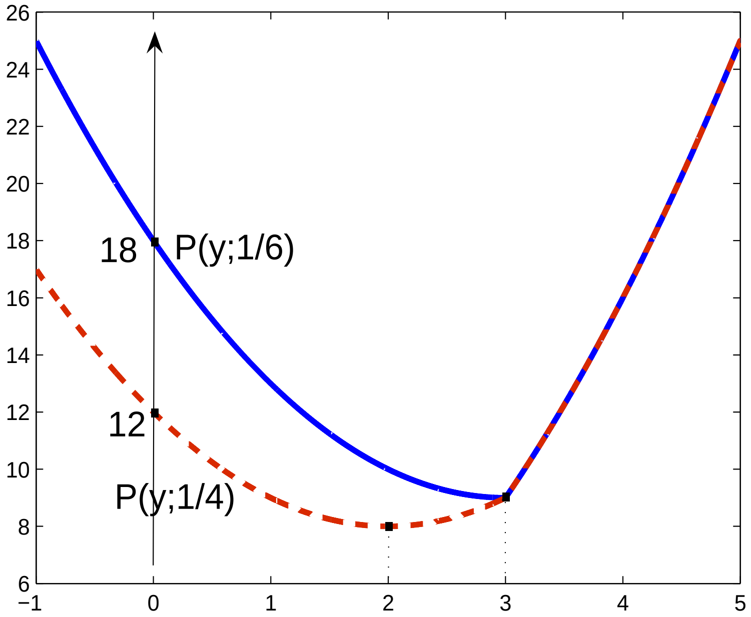

Figure 9: The convex nondifferentiable penalty function $P(y;\varepsilon) = \{y^2 + 1/\varepsilon \max\{0, 3-y\}\}$ for problem (29), with $\varepsilon = 1/4 > 1/6$ (*dashed line*) and $\varepsilon = 1/6$ (*continuous line*). By loosely solving (30) with $\Omega = \mathbb{R}$, for both the choices of ε, turns to be harmful when $\varepsilon = 1/4$, since clearly the solution $y^* = 3$ cannot be found.

$\Omega \equiv \mathbb{R}^n$ (so that $\overset{\circ}{\Omega} \supset \mathcal{F} = \{y \in \mathbb{R}^n : y \geq 3\}$), when tackling the (unconstrained) problem (30), one of the two choices of ε in Figure 9 (*dashed line*) yields a wrong value of the solution (i.e. the infeasible value $y^* = 2$ in place of $y^* = 3$). That is because the value of ε chosen is above the threshold 1/6, which is easily proved to be the maximum value of the penalty parameter which allows to find the unique stationary point $y^* = 3$ of (29).

On summary, the above considerations reveal the following couple of conclusions, which are essential to motivate our proposal for adaptively update the penalty parameters:

(1) the choice of Ω in (26) is essential to guarantee convergence towards local/global solutions of (25); moreover, the right choice of Ω is not known a priori. (E.g., starting the minimization of the penalty function in (30) from $\bar{y} = 5/2$, in case $\varepsilon = 1/4$ there might be the chance not to find the point $y^* = 3$);

(2) the choice of ε in (26) is crucial. (E.g., setting $\varepsilon > 1/6$ in (30), by no means a descent method will allow to find the unique stationary point $y^* = 3$, regardless of the choice of Ω).

On the other hand, to overcome the drawbacks (1) and (2) we might be induced to set both $\Omega = \mathbb{R}^n$ and ε very small. However, the latter choice on one hand may still suffer from (2); moreover, when ε is too small, serious ill-conditioning may arise, implying numerical instability and possibly slow progress.

For our purposes, since the focus of this paper is not on evaluating different penalty approaches to solve our constrained portfolio problem, we decided to adopt a very standard exact penalty framework from the literature. Indeed, here the proposal in [23] is considered, for its simplicity and since it guarantees sufficient theoretical results under mild assumptions. Nevertheless, we cannot exclude that a more complete investigation might be carried on, in a separate paper, exploiting different and specific penalty approaches, along with their relative properties. Given problem (25), the ℓ_1-nondifferentiable exact penalty function we adopt in this paper is therefore given by

$$P(y;\varepsilon) = f(y) + \frac{1}{\varepsilon} \left\| \max\{0, g(y)\}, h(y) \right\|_1, \tag{31}$$

which satisfies the next proposition (see also [24]).

Proposition 2 *Consider the problem (25) and assume that*

- *$\mathcal{F}$ is compact;*
- *MFCQ holds at any global minimum of (25);*
- *the set Ω in (26) satisfies $\overset{\circ}{\Omega} \supset \mathcal{F}$;*
- *the function $P(y;\varepsilon)$ in (26), with $\varepsilon \in \mathbb{R}$ and $\varepsilon > 0$, is defined as in (31).*

Then, there exists a value $\varepsilon^* > 0$ such that for any $\varepsilon \in (0, \varepsilon^*]$, any global minimum of (25) is a global minimum of (26) and viceversa.

The function $P(x;\varepsilon)$ satisfying the latter proposition is often addressed in the literature as a *weakly exact nondifferentiable penalty function.* A stronger result with respect to Proposition 2, involving also local minima, can be proved, provided that a stronger qualification condition is assumed. Since we are only concerned with tackling the global optima of (25), the results in Proposition 2 are sufficient to our purposes. Note that for our portfolio problem (20), after a brief computation the condition (a) of Definition 3 is always satisfied, at any feasible point. Furthermore, as we similarly discussed in [2], the alternative proposals in [6,28] strongly differ from our approach.

In our framework we adopt the nondifferentiable exact penalty function in (31), where we try to cope with the drawbacks (1) and (2) by setting $\Omega = \mathbb{R}^n$ and adaptively updating the parameter ε (see also [8]). More in particular (see also Section 6.1) we introduce the vector of parameters $\varepsilon \in \mathbb{R}^8$, and update it periodically, depending on the progress of the iterative PSO procedure. The latter expedient is also suggested by the fact that (see [2]) we use PSO to solve (26), so that some standard convergence properties of exact iterative methods are partially lost, and an additional strategy to steer optimization is expected in order to improve performance. Considering (20), the resulting ℓ_1-penalty problem we solve becomes

$$\min_{x \in \mathbb{R}^N,\ z \in \mathbb{R}^N} \ P(x,z;\varepsilon), \qquad \varepsilon \in \mathbb{R}^8, \tag{32}$$

where

$$\begin{aligned} P(x,z;\varepsilon) \ &= \ \rho_{a,p}(r) + \frac{1}{\varepsilon_0} \Bigg[\varepsilon_1 \max\left\{0, r_e - \sum_{i=1}^N \hat{r}_i x_i \right\} + \varepsilon_2 \left| \sum_{i=1}^N x_i - 1 \right| + \varepsilon_3 \max\left\{0, K_d - \sum_{i=1}^N z_i \right\} \\ &\qquad + \varepsilon_4 \max\left\{0, \sum_{i=1}^N z_i - K_u \right\} + \varepsilon_5 \sum_{i=1}^N \max\left\{0, z_i d - x_i \right\} \\ &\qquad + \varepsilon_6 \sum_{i=1}^N \max\left\{0, x_i - z_i u \right\} + \varepsilon_7 \sum_{i=1}^N \left| z_i(1 - z_i) \right| \Bigg] \end{aligned}$$

and $\varepsilon = (\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_7)^T > 0$, so that the entries of the penalty vector ε are adaptively updated at PSO iteration k when solving (32), as detailed in Section 6.1.

Observe that the adaptive change of the penalty parameter vector $\varepsilon \in \mathbb{R}^8$ has become more recently a common choice in the literature, and does not represent definitely a novel contribution of this paper. Moreover, by (32) observe that since $\rho_{a,p}(r)$ is convex in $\mathbb{R}^n$, it is easily seen that $P(x,z;\varepsilon)$ is convex with respect to the subvector x. As a consequence, its minimization is expected to possibly gain advantage of the latter structure and rapidly assess the first N entries (i.e. those referred to subvector x) of a solution. Conversely, the feasible set of (20) is surely compact, due to the bound constraints on the subvector x, but the existence of a unique minimizer is not guaranteed. Hence, the necessity to tackle the problem (32) by a global method. Finally, considering that the formulation (32) is NP-hard, and tradesmen/consultants often need a fast approximate solution of their portfolio problems, in order to compare different scenarios, we provisionally decided to discard asymptotically convergent exact global methods, when solving (32). In this regard, our choice of adopting PSO seems to provide a reasonable compromise between precision of the approximate solution in the early iterations, and computational burden, as the numerical results in Section 7 seem to confirm.

9 An equivalent model used as benchmark

In this section, we reformulate problem (20) as a nonlinear mixed-integer programming problem. This reformulation provides a model whose instances can be exactly solved by standard algorithms as FilMINT (which is an iterative method based on a branch-and-cut framework), publicly available on NEOS server (http://www.neos-server.org/neos/). The main purpose of this section is to provide a formulation of our portfolio selection problems, whose exact solution can be possibly used as a reference in our numerical experience (see Tables 2-3).

In accordance with [16], the resulting nonlinear mixed-integer programming problem with $2N + 2T$ unknowns is:

$$\min_{x,z,v,w} \quad \frac{a}{T}\sum_{t=1}^{T} v_t + \frac{1-a}{T^{\frac{1}{p}}}\left(\sum_{t=1}^{T} w_t^p\right)^{\frac{1}{p}} - \sum_{i=1}^{N} \hat{r}_i x_i \tag{33a}$$

$$v_t - w_t = \sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i, \qquad t = 1, \dots, T \tag{33b}$$

$$\sum_{i=1}^{N} \hat{r}_i x_i \geq r_e \tag{33c}$$

$$\sum_{i=1}^{N} x_i = 1 \tag{33d}$$

$$K_d \leq \sum_{i=1}^{N} z_i \leq K_u \tag{33e}$$

$$d z_i \leq x_i \leq u z_i, \qquad i = 1, \dots, N \tag{33f}$$

$$z_i \in \{0, 1\}, \qquad i = 1, \dots, N \tag{33g}$$

$$v_t, w_t \geq 0, \qquad t = 1, \dots, T \tag{33h}$$

In the above formulation, conditions (33c)-(33g) are trivially equivalent to (20b)-(20f). In addition, we write the objective function (20a) as in (22) where we introduce the variables $v^T = (v_1, \dots, v_T)$, $w^T = (w_1, \dots, w_T)$ as a standard trick to linearize the terms $\left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_{i,t})x_i\right)^+$ and $\left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_{i,t})x_i\right)^-$. Indeed, conditions (33b) and (33h) imply that

$$v_t \geq \left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_{i,t})x_i\right)^+ \quad \text{and} \quad w_t \geq \left(\sum_{i=1}^{N}(r_{i,t} - \hat{r}_{i,t})x_i\right)^- \quad t = 1, \dots, T. \tag{34}$$

In particular, we can rewrite (33b) as $v_t = \sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i + w_t$. Hence, as $w_t \geq 0$, we obtain $v_t \geq \sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i$ and, as $v_t \geq 0$, $v_t \geq \max\{\sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)x_i, 0\}$. We finally observe that the inequalities (34) hold as equalities in any minimum of problem (33). As an example consider any feasible solution

$(\bar{x}, \bar{z}, \bar{v}, \bar{w})$, for which the value of $\sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)\hat{x}_i$ is nonnegative and such that $\bar{v}_t > \sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)\bar{x}_i$ and $\bar{w}_t > 0$. Since variables v and w have positive coefficients in the objective function (33a) and no other constraint different from (33b) and (33h) involves v and w, then it is immediate to find feasible solution $(\bar{x}, \bar{z}, \tilde{v}, \tilde{w})$ that dominates $(\bar{x}, \bar{z}, \bar{v}, \bar{w})$. Just set $\tilde{v} = \bar{v}$, respectively $\tilde{w} = \bar{w}$, except for $\tilde{v}_t = \sum_{i=1}^{N}(r_{i,t} - \hat{r}_i)\bar{x}_i$, respectively for $\tilde{w}_t = 0$.

As indicated in Section 6, we use formulation (33) to obtain the optimal values ρ^* of our portfolio selection problem instances, to be compared with the values obtained through PSO. Unfortunately, the optimal solution of problem (33) may be particularly cumbersome to compute by an exact solver, when a large size instance is considered. All the same, we can use the linear relaxation of (33) to easily obtain a lower bound of ρ^* and, hence, to be able to asses the PSO performances even for a large size instance.

Indeed, we can observe that, if we linearly relax constraints (33g) with $0 \leq z_i \leq 1$, the objective function (33a) becomes a convex functional as it is sublinear. In addition, also the solution feasible set defined by constrains (33b)-(33h) becomes convex. Consequently, all the feasible stationary points of the linearly relaxed version of (33) are optimal and the value assumed by the objective function (33a) in these points provides a lower bound for ρ^*. Due to the convexity properties of the relaxed version of (33a), its stationary points are relatively easy to determine using standard nonlinear optimization algorithms.

10 Conclusions and future work

In this paper, first we have proposed a novel deterministic method for the initialization of the positions and the velocities of the swarm particles. Then, we have performed a numerical experience, using as optimization problem the unconstrained reformulation of a realistic complex portfolio selection problem. Finally, in order to effectively tune the parameters characterizing the latter unconstrained reformulation, we have proposed two different frameworks, where our new PSO initialization is tested. We both reported theoretical achievements and obtained satisfactory numerical results.

Nevertheless, our proposal seems to offer further opportunities for possible improvements and extensions. In particular:

- As stated in Section 1, regarding the unconstrained reformulation of our portfolio selection problems, we point out that other type of reformulations may be considered, both smooth and nonsmooth. To this purpose, resorting to exact penalty functions appears particularly promising, which encourages us to further propose our PSO initialization.
- According with the theory in Sections 3-4, we are persuaded that some improvements to our proposal may possibly come by monitoring the fading of the free response associated to any particle trajectory, in the extended space $\mathbb{R}^{2n}$.
- Combining PSO with a globally convergent method for derivative-free optimization may possibly provide a better quality of the solution. The latter fact deserves further investigation, in order to further exploit the structure of portfolio selection problems (i.e. the convexity of problem (20) with respect to the subvector of unknowns x).

References

[1] G. Campana, E.F.and Fasano and A. Pinto, “Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization,” *Journal of Global Optimization*, vol. 48, pp. 347–397, 2010.

[2] M. Corazza, G. Fasano, and R. Gusso, “Particle Swarm Optimization with non-smooth penalty reformulation, for a complex portfolio selection problem,” *Applied Mathematics and Computation*, vol. 224, pp. 611–624, 2013.

[3] V. Nannen and A. E. Eiben, “Relevance estimation and value calibration of evolutionary algorithm parameters,” in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI’07, (San Francisco, CA, USA), pp. 975–980, Morgan Kaufmann Publishers Inc., 2007.

[4] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia*, IEEE Service Center, Piscataway, NJ, IV, 1995.

[5] Y. Zhang, S. Wang, and G. Ji, “A comprehensive survey on particle swarm optimization algorithm and its applications,” *Mathematical Problems in Engineering*, Article ID 931256, 2015.

[6] D. Rini, S. Shamsuddin, and S. Yuhaniz, “Particle swarm optimization: Technique, system and challenges,” *International Journal of Computer Applications*, vol. 14, pp. 19–27, 2011.

[7] E. Campana, G. Fasano, and D. Peri, “Penalty function approaches for ship multidisciplinary design optimisation (MDO),” *European Journal of Industrial Engineering*, vol. 6, pp. 765–784, 2012.

[8] B. Qiao, Y. Chang, M. Cui, and K. Yao, “Hybrid particle swarm algorithm for solving nonlinear constraint optimization problems,” *WSEAS Transactions on Mathematics*, vol. 12, pp. 76–84, 2013.

[9] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimisation: an overview,” *Swarm Intelligence Journal*, vol. 1, pp. 33–57, 2007.

[10] C. Cleghorn and A. Engelbrecht, “Particle swarm variants: standardized convergence analysis,” *Swarm Intelligence*, vol. Online First, pp. 1–27, 2015.

[11] E. Campana, M. Diez, G. Fasano, and D. Peri, *Springer Lecture Notes in Computer Science*, ch. Improving the initial particles position and parameters selection for PSO, in Bound Constrained Optimization problems, pp. 112–119. Springer-Verlag Berlin Heidelberg, D, 2013.

[12] T. Kolda, R. Lewis, and V. Torczon, “Optimization by direct search: New perspectives on some classical and modern methods,” *SIAM Review*, vol. 45, pp. 385–482, 2003.

[13] M. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–435, 1952.

[14] G. Fasano and M. Roma, “Iterative computation of negative curvature directions in large scale optimization,” *Computational Optimization and Applications*, vol. 38, pp. 81–104, 2007.

[15] G. Fasano and S. Lucidi, “A nonmonotone truncated Newton-Krylov method exploiting negative curvature directions, for large scale unconstrained optimization,” *Optimization Letters*, vol. 3, pp. 521–535, 2009.

[16] Z. Chen and Y. Wang, “Two-sided coherent risk measures and their application in realistic portfolio optimization,” *Journal of Banking & Finance*, vol. 32, pp. 2667–2673, 2008.

[17] J. Nocedal and S. Wright, *Numerical Optimization (Springer Series in Operations Research and Financial Engineering) - Second Edition.* New York: Springer, 2000.

[18] A. Schaerf, “Local search techniques for constrained portfolio selection problems,” *Computational Economics*, vol. 20, pp. 177–190, 2002.

[19] V. Nannen and A. E. Eiben, "Relevance Estimation and Value Calibration of Evolutionary Algorithm Parameters," in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence* (M. M. Veloso, ed.), pp. 1034–1039, 2007.

[20] E. Montero, M. Riff, and B. Neveu, "A beginner's guide to tuning methods," *Appl. Soft Comput.*, vol. 17, pp. 39–51, 2014.

[21] A. E. Eiben and J. Smith, *Introduction to Evolutionary Computing.* Springer Verlag, 2003.

[22] M. Gilli and E. Kllezi, "The threshold accepting heuristic for index tracking," in *Financial Engineering, E-commerce and Supply Chain* (P. Pardalos and V. Tsitsiringos, eds.), vol. 70 of *Applied Optimization*, pp. 1–18, Springer US, 2002.

[23] W. Zangwill, "Nonlinear programming with penalty functions," *Management Science*, vol. 13, pp. 344–358, 1967.

[24] O. Mangasarian and S. Han, "Exact penalty functions in nonlinear programming," *Mathematical Programming*, vol. 17, pp. 251–269, 1979.