On the Computation of All Global Minimizers Through Particle Swarm Optimization

Konstantinos E. Parsopoulos and Michael N. Vrahatis

***Abstract*—This paper presents approaches for effectively computing all global minimizers of an objective function. The approaches include transformations of the objective function through the recently proposed deflection and stretching techniques, as well as a repulsion source at each detected minimizer. The aforementioned techniques are incorporated in the context of the particle swarm optimization (PSO) method, resulting in an efficient algorithm which has the ability to avoid previously detected solutions and, thus, detect all global minimizers of a function. Experimental results on benchmark problems originating from the fields of global optimization, dynamical systems, and game theory, are reported, and conclusions are derived.**

***Index Terms*—Deflection technique, detecting all minimizers, dynamical systems, Nash equilibria, particle swarm optimization (PSO), periodic orbits, stretching technique.**

I. Introduction

THE MINIMIZATION of multimodal functions with numerous local and global minima is a problem that frequently arises in diverse scientific fields. This problem is NP-hard in the sense of its computational complexity even in simple cases [1]. However, in many cases, the value of the global minimum of the objective function is *a priori* known, due to the form of the function (quadratic, nonnegative, etc.). For example, nonlinear least squares problems, as well as feedforward neural network training are such problems, where the involved objective functions have *a priori* known global minimum, equal to zero.

In general, the nature of some applications is such that it is necessary to detect not just one, but all the global minimizers. Such an example is the *computation of Nash equilibria* in game theory. Nash equilibria can be considered as steady state solutions of a game. In a Nash equilibrium, each player selects the optimal strategy given the strategies of all other players, i.e., the strategy that yields the largest payoff. However, different Nash equilibria correspond to significantly different outcomes of the game. It is, therefore, necessary to compute not just one but all the Nash equilibria in order to produce a reliable estimate of the outcome that can be reached through playing a game.

An interesting application which requires the computation of more than one global minimizer, is the *computation of periodic orbits of nonlinear mappings*. Nonlinear mappings are widely used to model conservative, or dissipative dynamical systems. A central role in the study of such mappings is played by points which remain invariant under the mapping. These points are called fixed points or periodic orbits of the mapping and they may be further categorized into several types (stable, unstable, etc.). Developing techniques for detecting all such points, or all points of a specific type, is an area of intense ongoing research.

The last three decades have witnessed the development of efficient and effective stochastic optimization algorithms. In contrast to the traditional adaptive stochastic search algorithms, evolutionary computation (EC) techniques exploit a set of potential solutions, named a *population*, and detect the optimal solution through cooperation and competition among the individuals of the population. These techniques often detect optima in difficult optimization problems faster than traditional optimization methods. The most frequently encountered population-based EC techniques, such as evolution strategies (ES) [2]–[7], genetic algorithms (GAs) [8], [9], genetic programming [10], [11], and evolutionary programming [12], are inspired from the evolutionary mechanisms of nature.

The particle swarm optimization (PSO) algorithm belongs to the category of Swarm Intelligence methods. It was developed and first introduced as a stochastic optimization algorithm by Eberhart and Kennedy [13]. During the last seven years, PSO gained increasing popularity due to its effectiveness in performing difficult optimization tasks. Among other applications, it has been applied to tackle multiobjective problems [14]–[19], minimax problems [20], [21], integer programming problems [22], noisy and continuously changing environments [23]–[26], ℓ_1 errors-in-variables problems [27], existence of function zeros [28], and numerous engineering applications [29]–[40].

This paper proposes a technique aiming to compute all global minimizers, while avoiding local minimizers, through PSO. This technique incorporates the recently proposed *deflection* and *stretching* procedures to alleviate local minimizers. Some of the problems that may arise when using these two techniques are overcome by incorporating a *repulsion* technique. The proposed approach can be used in combination with PSO to detect all global minimizers effectively. The performance of the algorithm is illustrated on test problems originating from various scientific fields, such as global optimization, dynamical systems, and game theory.

The rest of this paper is organized as follows. Section II is devoted to the exposition of the PSO method and its variants. In Section III, the proposed technique, as well as other established approaches are described and analyzed. In Section IV, test problems from global optimization, dynamical systems, and game theory are described, and experimental results are reported. The paper closes with conclusions and ideas for further research in Section V.

Manuscript received July 10, 2002; revised September 10, 2003. This work was supported in part by the Pythagoras and PENED 2001 Research Grants awarded by the Greek Ministry of Education and Religious Affairs, the General Secretariat for Research and Technology, and by the European Union.

K. E. Parsopoulos is with the Department of Mathematics, University of Patras, Patras GR-26110, Greece (e-mail: kostasp@math.upatras.gr).

M. N. Vrahatis is with the Department of Mathematics, University of Patras, Patras GR-26110, Greece (e-mail: vrahatis@math.upatras.gr).

Digital Object Identifier 10.1109/TEVC.2004.826076

II. Particle Swarm Optimization (PSO) Algorithm

PSO belongs to the broad class of stochastic optimization algorithms. The ideas that underlie PSO are inspired not by the evolutionary mechanisms encountered in natural selection, but rather by the social behavior of flocking organisms, such as swarms of birds and fish schools. It has been observed that the behavior of the individuals that comprise a flock adheres to fundamental rules like nearest-neighbor velocity matching and acceleration by distance [41], [42]. In this respect, it has been claimed that PSO performs mutation with a conscience [43].

PSO is a population-based algorithm that exploits a population of individuals to probe promising regions of the search space. In this context, the population is called a *swarm* and the individuals are called *particles*. Each particle moves with an adaptable velocity within the search space, and retains in its memory the best position it ever encountered. In the *global* variant of PSO the best position ever attained by all individuals of the swarm is communicated to all the particles. In the *local* variant, each particle is assigned to a neighborhood consisting of a prespecified number of particles. In this case, the best position ever attained by the particles that comprise the neighborhood is communicated among them [41]. This paper considers the global variant of PSO only.

Assume an n-dimensional search space, $S \subset \mathbb{R}^n$, and a swarm consisting of N particles. The ith particle is in effect an n-dimensional vector

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{in})^\top \in S.$$

The velocity of this particle is also an n-dimensional vector

$$V_i = (v_{i1}, v_{i2}, \ldots, v_{in})^\top \in S.$$

The best previous position encountered by the ith particle is a point in S, denoted as

$$P_i = (p_{i1}, p_{i2}, \ldots, p_{in})^\top.$$

Assume g to be the index of the particle that attained the best previous position among all the individuals of the swarm, and t to be the iteration counter. Then, according to the first version of PSO, as introduced in the pioneering work of Kennedy and Eberhart [42], the swarm is manipulated according to the following equations:

$$V_i(t+1) = V_i(t) + cr_1\left(P_i(t) - X_i(t)\right) + cr_2\left(P_g(t) - X_i(t)\right) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

where $i = 1, 2, \ldots, N$ is the particle's index, c is a positive constant, called the *acceleration constant*, r_1, r_2 are random numbers, uniformly distributed within the interval [0,1], and $t = 1, 2, \ldots,$ indicates the iterations.

A drawback of the aforementioned version of PSO is associated with the lack of a mechanism responsible for the control of the magnitude of the velocities, which fosters the danger of swarm explosion and divergence [44]. To address the explosion problem a threshold $V_{\max}$ on the absolute value of the velocity that can be assumed by any particle was incorporated. This modification alone, however, proved inadequate in significantly enhancing the effectiveness of the algorithm. A closer inspection of the operation of the algorithm indicated that although PSO located the region of the optimum faster than different EC algorithms, once in this region the algorithm progressed slowly, due to the inability to adjust the velocity stepsize to continue the search at a finer grain [44].

The aforementioned problem was addressed by further incorporating a weight parameter on the previous velocity of the particle. The resulting equations for the manipulation of the swarm are [45]–[47]

$$V_i(t+1) = wV_i(t) + c_1r_1\left(P_i(t) - X_i(t)\right) + c_2r_2\left(P_g(t) - X_i(t)\right) \quad (3)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (4)$$

where $i = 1, 2, \ldots, N$, w is a parameter called the *inertia weight*, c_1 and c_2 are positive constants, referred to as *cognitive* and *social* parameters, respectively, and r_1, r_2 are random numbers, uniformly distributed in [0,1].

The inertia weight w in (3), is employed to manipulate the impact of the previous history of velocities on the current velocity. Therefore, w resolves the tradeoff between the global (wide ranging) and local (nearby) exploration ability of the swarm. A large inertia weight encourages global exploration (moving to previously not encountered areas of the search space), while a small one promotes local exploration, i.e., fine-tuning the current search area. A suitable value for w provides the desired balance between the global and local exploration ability of the swarm and, consequently, improves the effectiveness of the algorithm. Experimental results suggest that it is preferable to initialize the inertia weight to a large value, giving priority to global exploration of the search space, and gradually decreasing w so as to obtain refined solutions [46], [47]. This finding is also intuitively very appealing. In conclusion, an initial value of w around 1 and a gradual decline toward 0 is considered a proper choice for w.

An alternative version of PSO incorporates a parameter called the *constriction factor* and the swarm is manipulated according to the equations [48]

$$V_i(t+1) = \chi\left(V_i(t) + c_1r_1\left(P_i(t) - X_i(t)\right) + c_2r_2\left(P_g(t) - X_i(t)\right)\right) \quad (5)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (6)$$

where $i = 1, 2, \ldots, N$, χ is the constriction factor, c_1 and c_2 denote the cognitive and social parameters, respectively, and r_1, r_2 are random numbers uniformly distributed in the interval [0,1]. The value of the constriction factor is typically obtained through the formula [48]

$$\chi = \frac{2\kappa}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|} \quad (7)$$

for $\phi > 4$, where $\phi = c_1 + c_2$, and $\kappa = 1$. Different configurations of χ, as well as a theoretical analysis of the derivation of (7), can be found in [48].

Proper fine-tuning of the parameters c_1 and c_2, results in better performance of PSO. An extended study of the acceleration parameter c, in the primary version of PSO, is provided in [49]. As default values, $c_1 = c_2 = 2$ were proposed, but experimental results indicate that alternative configurations, depending on the problem at hand, may produce superior performance.

The initialization of the swarm and the velocities, is usually performed randomly in the search space, although more sophisticated initialization techniques can enhance the overall performance of the algorithm [50]. For uniform random initialization

in a multidimensional search space, a Sobol sequence generator can be used [51].

Thorough theoretical investigations of the convergence properties of PSO are provided in [48] and [52]. These studies first consider a simplified deterministic model of the algorithm in order to provide an understanding about how it probes the search space, and then proceed to analyze the full stochastic system [48], [52]. Generalized models of the algorithm are proposed, and techniques for controlling the convergence properties of the particle system by fine-tuning its parameters are analyzed in [48] and [52].

III. Established Approaches and the Proposed Techniques

Various techniques reported in the literature can be incorporated in the context of optimization algorithms in order to obtain several minimizers of a function [53], [54]. A simple approach is the *multistart* technique. Once the optimization algorithm has converged to a minimizer, this technique reinitializes the algorithm at a random point of the search space. On the other hand, this approach includes no mechanisms to deter convergence toward previously detected minimizers [54].

Alternatively, the detection of more than one minimizer can be achieved by modifying the objective function, so as to contain information concerning the position of the previously detected minimizers, in its new form. In this context, Goldstein and Price proposed an efficient algorithm for the minimization of algebraic functions, which exploits higher order derivatives of the involved polynomials [55]. The technique was later generalized for nonpolynomial problems using a transformation, which involves the Hessian of the objective function. However, the numerical computation of the Hessian is not always feasible and in any case, it is computationally expensive. Thus, in its general form, this approach is not widely applicable. Similar algorithms were proposed by Shusterman [56], but, after the detection of a few minimizers, the objective function becomes very flat and, thus, minimization is becoming increasingly difficult. An interesting approach is the *tunneling* technique, proposed by Vilkov *et al.* [57] for one-dimensional (1-D) functions, and generalized for the multidimensional case by Gomez and Levy [58], and Montalvo [59]. However, the hypersurface constructed by the tunneling algorithm becomes very flat as the number of the detected minimizers increases, hindering further exploration of the search space, after the detection of a few minimizers. A different technique for avoiding local minimizers is *simulated annealing* that combines local search with Monte Carlo techniques and simulates the annealing processes which are used to reveal the low temperature state of materials [60].

Filled functions are another widely used approach for avoiding local minimizers, developed by Ge [61], [62]. This technique also requires a transformation of the given objective function, according to the following equation:

$$T(x;r,p) = f(x) + \frac{\alpha}{r + f(x)} \exp\left(-\frac{\|x - x^*\|^2}{p^2}\right) \tag{8}$$

where x^* is a detected minimizer and α, r, p are arbitrary parameters. The first multiplier of (8) inverts the objective function, and turns the local minimum at x^* to a local maximum, while the second multiplier imposes a penalty when approaching x^*. The magnitude of the penalty is measured in units of p^2. The parameter α controls the extent of the impact of the transformation on f.

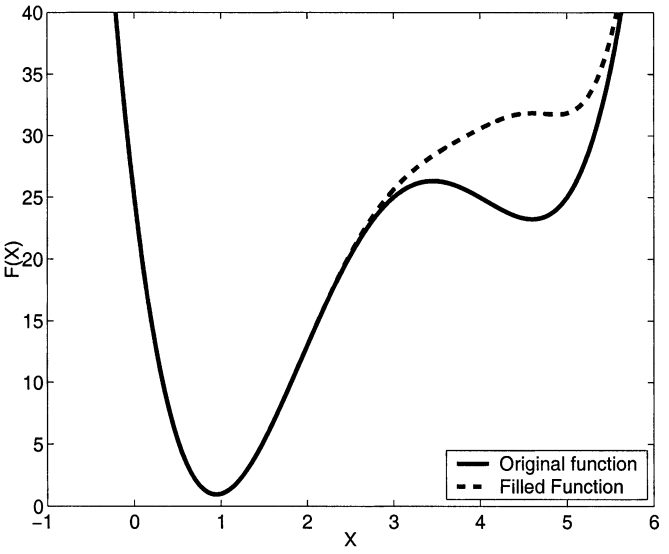

Fig. 1. Filled function transformation of the function defined in (9), at point $x^* = 4.600\,955\,89$, for the parameter values $\alpha = 200$, $r = 0$, and $p = 1$.

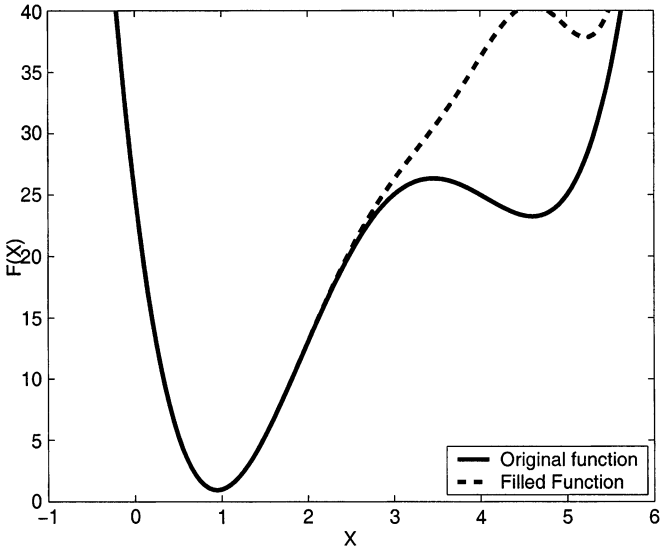

Fig. 2. Filled function transformation of the function defined in (9), at point $x^* = 4.600\,955\,89$, for the parameter values $\alpha = 400$, $r = 0$, and $p = 1$.

In Figs. 1 and 2, the effect of the filled function transformation is illustrated for the 1-D function

$$f(x) = x^4 - 12x^3 + 47x^2 - 60x + 25 \tag{9}$$

on the local minimizer $x^* = 4.600\,955\,89$. The figures were obtained for the fixed values $r = 0$ and $p = 1$, and for two different values of α, $\alpha = 200$ and $\alpha = 400$, respectively.

The filled function transformation introduces new local minima at both sides of the detected minimizer. The new local minimizers have higher function values than the detected one and, thus, form a shape similar to a "mexican hat" around it, causing optimization algorithms to get trapped to these local minimizers.

Alternative filled functions were proposed by Ge and Qin [63], but it is not always evident which class of problems these techniques can be applied [54].

"*Deflection*" and "*stretching*" are two recently proposed techniques that also rely on the concept of transforming the objective function in a way that knowledge of previously detected minimizers is incorporated in its new form. Preliminary experimental results indicate that both of these techniques can be used in the context of PSO to alleviate the problem of introducing local minima and detect several global minimizers effectively [64], [65]. Difficulties that may arise using these approaches, may be overcome through a "*repulsion*" technique. In the following sections, the workings of the deflection, stretching, and repulsion techniques are exposed and analyzed.

A. Deflection Technique

Let

$$f : S \to \mathbb{R}, \quad S \subset \mathbb{R}^n$$

be the original objective function under consideration. Let

$$x_i^*, \quad i = 1, \ldots, m$$

be m minimizers of f. Then, the *deflection* technique is defined as

$$F(x) = T_1(x; x_1^*, \lambda_1)^{-1} \cdots T_m(x; x_m^*, \lambda_m)^{-1} f(x) \quad (10)$$

where λ_i, $i = 1, \ldots, m$ are relaxation parameters, and $T_1, \ldots, T_m$ are appropriate functions in the sense that the resulting function F has exactly the same minimizers as f, except at points $x_1^*, \ldots, x_m^*$. In other words, the functions $T_1, \ldots, T_m$, must be selected in such a way that any sequence of points $\{x_k\}_{k=0}^{\infty}$ converging to any one of the minimizers x_i^*, does not produce a minimum of F at $x = x_i^*$, while all other minima of f remain unaffected. The functions

$$T_i(x; x_i^*, \lambda_i) = \tanh(\lambda_i \|x - x_i^*\|), \quad i = 1, \ldots, m \quad (11)$$

satisfy the aforementioned property, known as the *deflection property*, as it is shown in [66].

The effect of the deflection procedure applied on the local minimizer $x^* = 4.600\,955\,89$ of the function defined by (9), is illustrated in Figs. 3–5.

The 1-D objective function

$$f(x) = \cos^2(x) + 0.1 \quad (12)$$

has the global minimum 0.1 at the global minimizers $x = k(\pi/2)$, with $k = \pm 1, \pm 2, \ldots$. There are no other minima (local or global). Suppose the minimizer $x^* = (\pi/2)$ has been detected, then the effect of deflection on f, for $\lambda = 1$, is exhibited in Fig. 6.

Alternative configurations of the parameter λ result in different shapes of the transformed function. The deflection effect for $\lambda = 10$ and $\lambda = 0.1$ is exhibited in Figs. 7 and 8, respectively. It is clear that for larger values of λ the effect of the deflection technique on the objective function is relatively mild. On the other hand, using $\lambda < 1$ results in a function F with considerably larger function values in the neighborhood of the deflected minimizer, and may even affect the value of neighboring minimizers, as exhibited in Fig. 8. In all figures, the deflection transformation introduces new local minima at both sides of the deflected minimizer, forming a "mexican hat" around the deflected point.

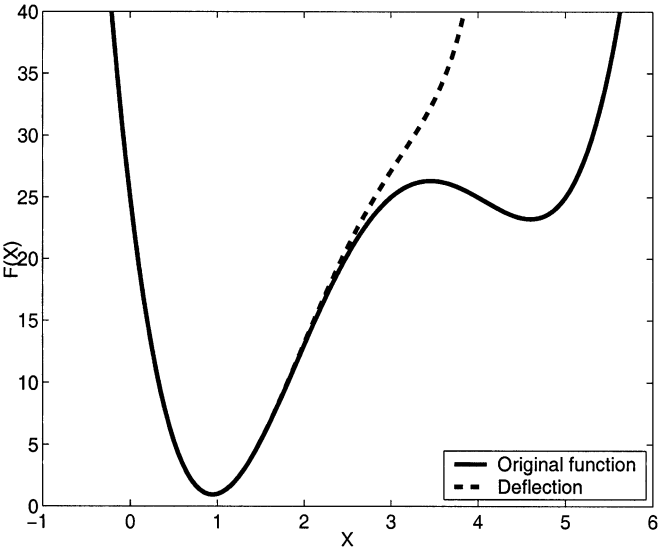

Fig. 3. Deflection transformation of the function defined in (9), at the point $x^* = 4.600\,955\,89$, for $\lambda = 1$.

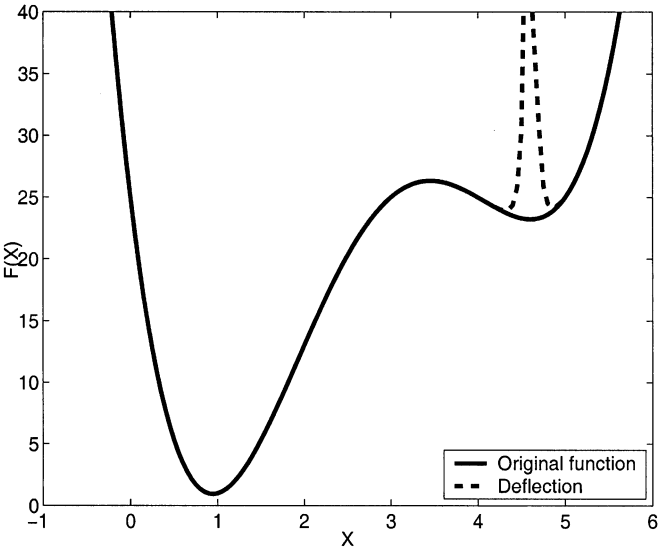

Fig. 4. Deflection transformation of the function defined in (9), at the point $x^* = 4.600\,955\,89$, for $\lambda = 10$.

A point to notice is that the deflection technique should not be used on its own on a function f, whose global minimum is zero. The reason is that the function F will also have zero value at the deflected global minimizer, since f will be equal to zero at such points. This problem can be easily alleviated by taking

$$\hat{f} = f + c$$

where $c > 0$ is a constant, instead of f. The function $\hat{f}$ possesses all the information regarding the minimizers of f, but the global minimum is increased from zero to c. Alternatively, the repulsion technique, which will be described later, can be used to overcome this problem.

In conclusion, the deflection technique can be used in cases where both global and local minimizers are required since it alleviates only the detected minimizers.

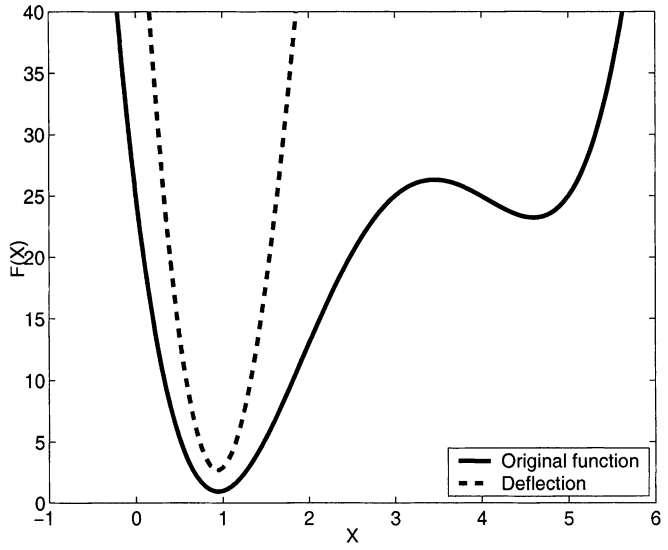

Fig. 5. Deflection transformation of the function defined in (9), at the point $x^* = 4.600\,955\,89$, for $\lambda = 0.1$.

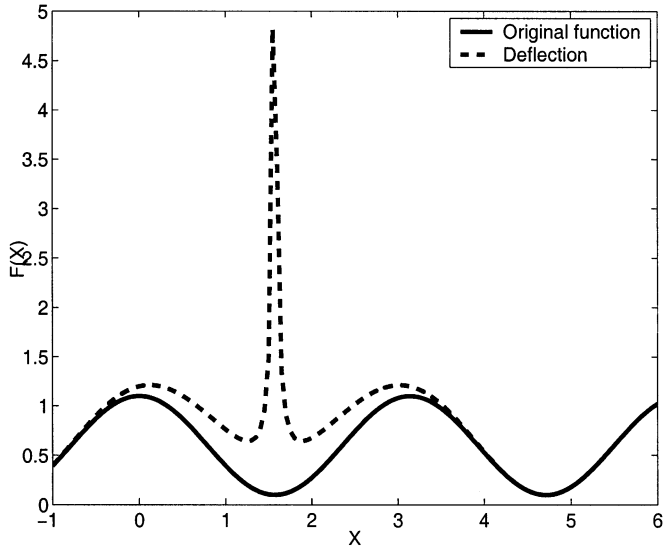

Fig. 6. Deflection transformation of the function defined in (12), at the point $x^* = (\pi/2)$, for $\lambda = 1$.

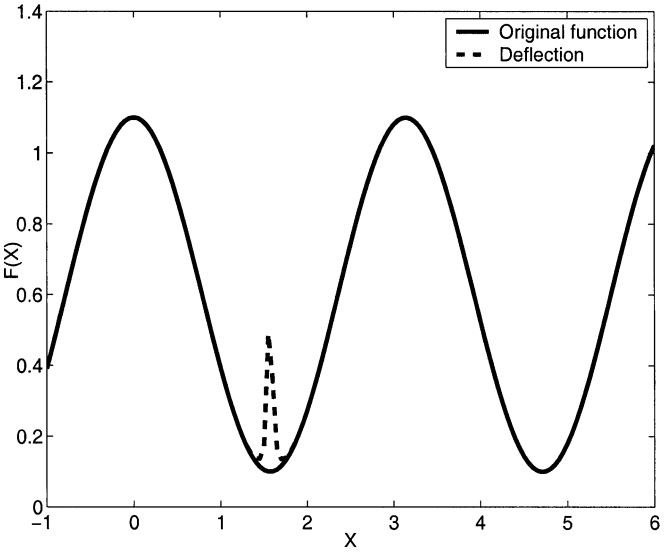

Fig. 7. Deflection transformation of the function defined in (12), at the point $x^* = (\pi/2)$, for $\lambda = 10$.

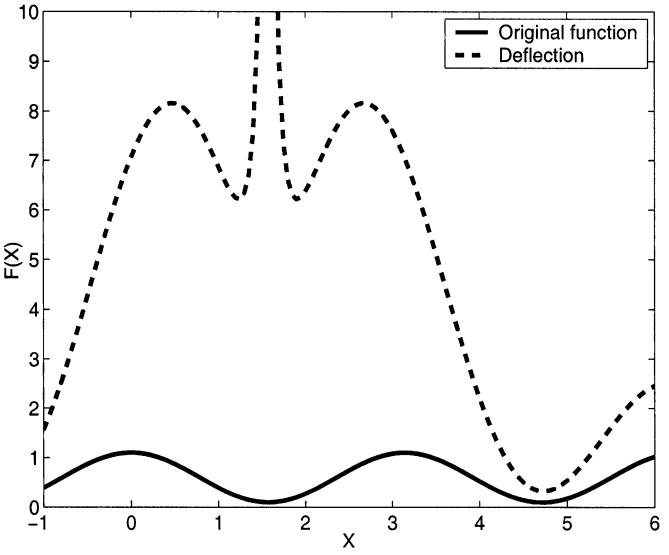

Fig. 8. Deflection transformation of the function defined in (12), at the point $x^* = (\pi/2)$, for $\lambda = 0.1$.

B. Stretching Technique

A different recently proposed technique, developed to address the problem of local minima, is the *stretching* technique [26], [64], [67], [68]. This technique consists of a two-phase transformation of the objective function. The first phase of the transformation, stretches the objective function upwards, eliminating all minima with values higher than the value of the obtained minimizer. In the second stage of the transformation, the detected minimum is turned to a maximum. All minima with lower values of the objective function remain unaltered by the transformation.

Let x^* be an obtained minimizer of the objective function f. Then, stretching is defined as [26], [64], [67], [68]

$$G(x) = f(x) + \gamma_1 \|x - x^*\| \left(\operatorname{sign}\left(f(x) - f(x^*)\right) + 1\right) \quad (13)$$

$$H(x) = G(x) + \gamma_2 \frac{\operatorname{sign}\left(f(x) - f(x^*)\right) + 1}{\tanh\left(\mu\left(G(x) - G(x^*)\right)\right)} \quad (14)$$

where γ_1, γ_2, and μ are arbitrary parameters. The $\operatorname{sign}(\cdot)$ function is the well known three-valued sign function, defined as

$$\operatorname{sign}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}.$$

The effect of the stretching transformation on the function defined in (9) is illustrated in Fig. 9, for $\gamma_1 = (3/2)$, $\gamma_2 = (1/2)$, and $\mu = 10^{-1}$. The parameter γ_1 controls the upward stretching of the objective function, performed by $G(x)$, in (13). Due to this transformation, the local minima with function values higher than the one found are eliminated. The effect of increasing the value of γ_1 in the aforementioned example, is exhibited in Fig. 10, for $\gamma_1 = (7/2)$. In practice, high values of γ_1 [e.g., $(10^4/2)$] are used in multidimensional problems to ensure that the local minima with function values higher than the detected one, will be eliminated.

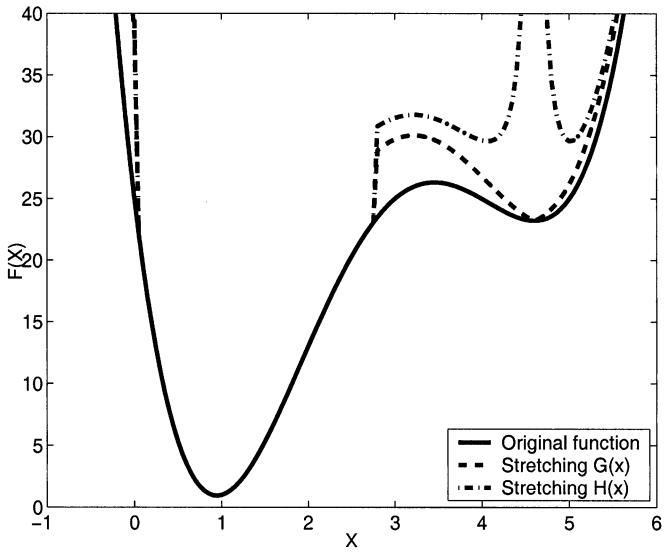

Fig. 9. Stretching transformations $G(x)$ (dashed line) and $H(x)$ (dash-dotted line) of the function defined in (9), at the point $x^* = 4.600\,955\,89$, for $\gamma_1 = (3/2)$, $\gamma_2 = (1/2)$, and $\mu = 10^{-1}$.

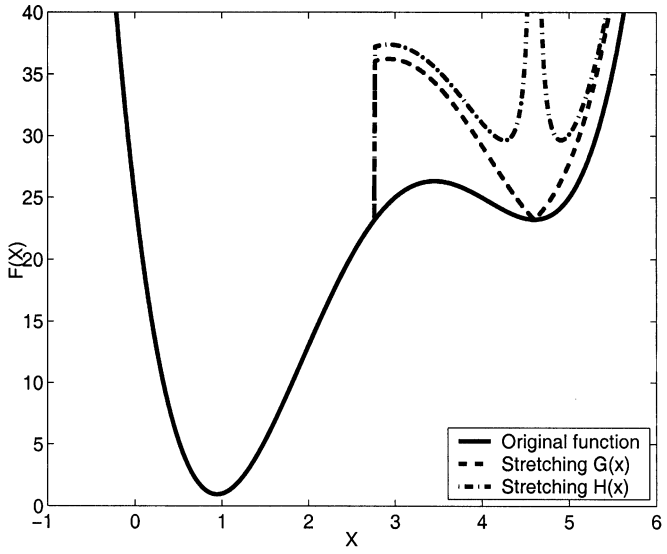

Fig. 10. Stretching transformations $G(x)$ (dashed line) and $H(x)$ (dash-dotted line) of the function defined in (9), at the point $x^* = 4.600\,955\,89$, for $\gamma_1 = (7/2)$, $\gamma_2 = (1/2)$, and $\mu = 10^{-1}$.

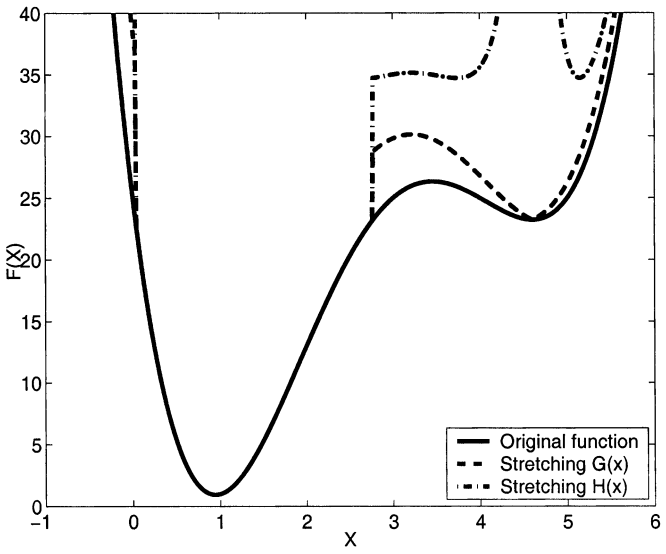

Fig. 11. Stretching transformations $G(x)$ (dashed line) and $H(x)$ (dash-dotted line) of the function defined in (9), at the point $x^* = 4.600\,955\,89$, for $\gamma_1 = (3/2)$, $\gamma_2 = (3/2)$, and $\mu = 10^{-1}$.

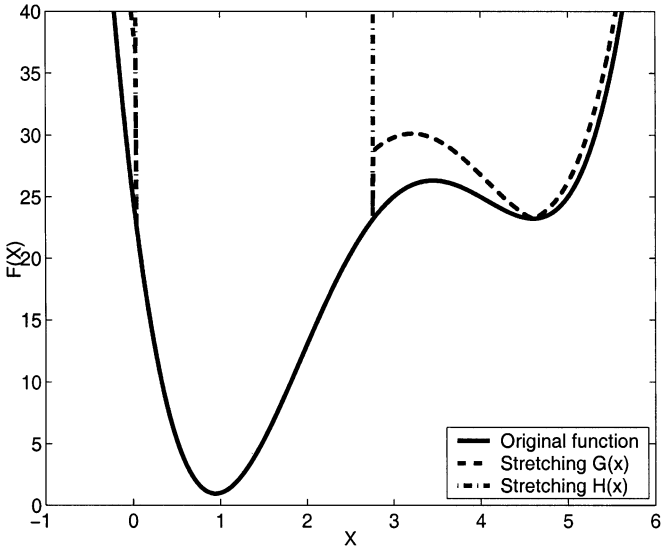

Fig. 12. Stretching transformations $G(x)$ (dashed line) and $H(x)$ (dash-dotted line) of the function defined in (9), at the point $x^* = 4.600\,955\,89$, for $\gamma_1 = (3/2)$, $\gamma_2 = (1/2)$, and $\mu = 10^{-2}$.

The parameters γ_2 and μ determine the range of the effect and the magnitude of the elevation, respectively. The effect of setting γ_2 to a larger value, $\gamma_2 = (3/2)$, is exhibited in Fig. 11. It is evident that a larger area around the local minimizer is affected by increasing this parameter. The effect of decreasing μ to 10^{-2} is illustrated in Fig. 12. Even slightly decreasing the parameter μ, the former local minimizer (and now local maximizer) takes extreme function values. Usually, $\gamma_1 = (10^4/2)$, $\gamma_2 = (1/2)$, and $\mu = 10^{-10}$ is a satisfactory setup, unless information for the objective function implying a different parameter setup is available.

The stretching transformation modifies neither local minima with function values lower than the obtained one nor global minima. It does, however, alleviate all minima with higher or equal function values. Thus, it is proper for local minima, but if it is applied on a global minimizer, then all other global minimizers are alleviated. This effect is displayed in Fig. 13, where the stretching transformation is applied on the global minimizer $x^* = (\pi/2)$ of the function defined in (12). It is clear that all other global minimizers of the objective function are alleviated.

A very interesting point to note is that the "mexican hat" effect which appears in the deflection technique, is also present in this setting. Proper fine-tuning of the stretching parameters tends to alleviate this problem, but for a given function, information concerning optimal parameter values or the appropriate parameter configuration to avoid this problem, is not generally available. If preprocessing for the determination of the proper parameters is not feasible or desirable, then a "*repulsion*" technique can be used to prevent the swarm from converging to one of the local minima artificially created due to the "mexican hat." A discussion of the "*repulsion*" technique is given in the next section.

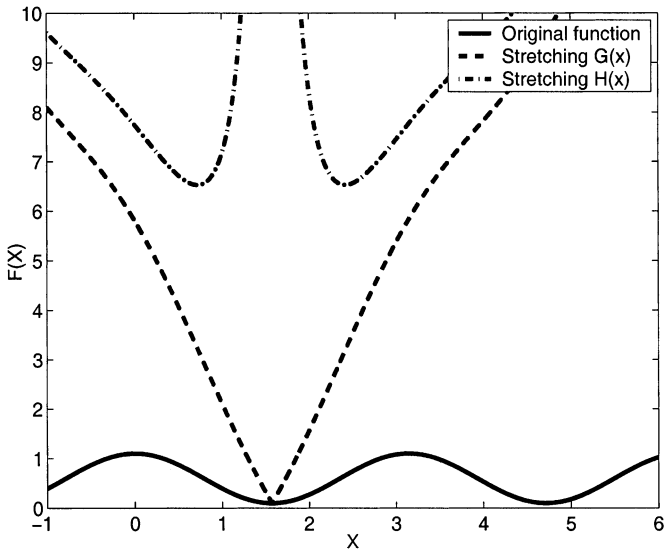

Fig. 13. Stretching transformations $G(x)$ (dashed line) and $H(x)$ (dash-dotted line) of the function defined in (12), at the point $x^* = (\pi/2)$, for $\gamma_1 = (3/2)$, $\gamma_2 = (1/2)$, and $\mu = 10^{-1}$.

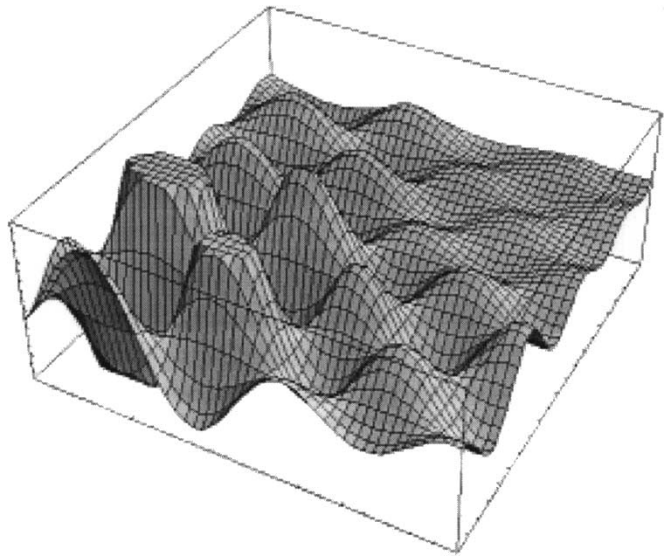

Fig. 14. Original plot of the *Levy no. 5* function in the range $[-2, 2]^2$.

An application of the stretching transformation on a two-dimensional (2-D) function, namely the *Levy no. 5* function, which is defined by

$$f(x) = \sum_{i=1}^{5} [i \cos((i-1)x_1 + i)] \times \sum_{j=1}^{5} [j \cos((j+1)x_2 + j)] + (x_1 + 1.425\,13)^2 + (x_2 + 0.800\,32)^2 \quad (15)$$

is illustrated in Figs. 14–16. Fig. 14 illustrates the original plot of the function. Fig. 15 illustrates the effect of stretching after the application of the first transformation $G(x)$, which is defined in (13), on a local minimizer at the upper part of the figure. All minima with lower function values are left unaffected. Finally, the second transformation $H(x)$, defined in (14), is illustrated in Fig. 16, where the stretched minimizer is transformed to a maximizer. The minima with lower function values have been left unaffected.

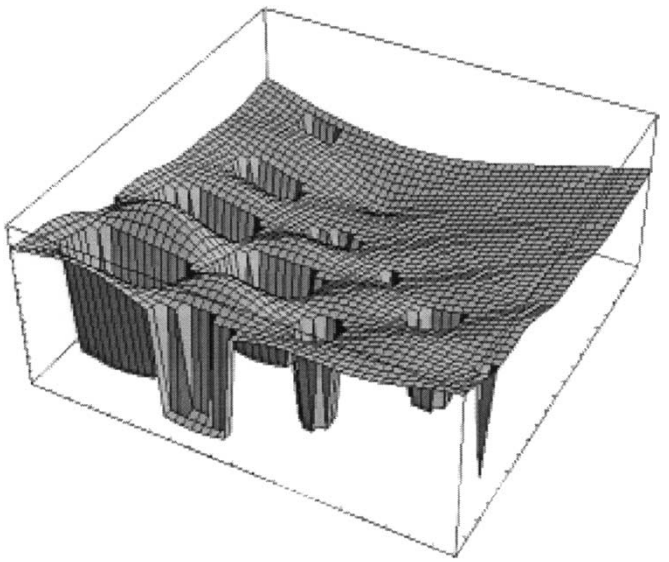

Fig. 15. First stage $G(x)$ of the stretching transformation (13) for the *Levy no. 5* function.

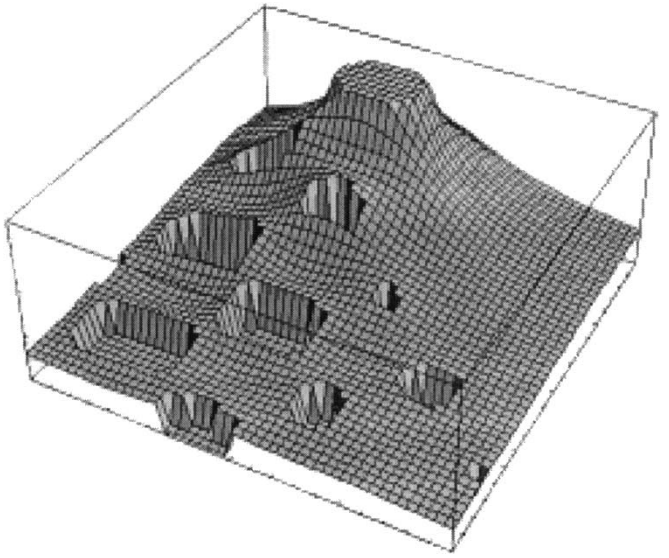

Fig. 16. *Levy no. 5* function after the stretching transformation (14).

In conclusion, the stretching technique can be used in cases where only the global minimizer is required since it alleviates the local ones.

C. Repulsion Technique

As previously mentioned, in numerous cases the transformations applied to eliminate minimizers of the original objective function f introduce new local minima in "mexican hat"-shaped areas of the resulting function. Thus, even after the application of the transformations, it is not certain that the swarm will not converge to a neighborhood of one of the already detected minimizers, since particles are solely guided by the function values of the positions they assume in the search space.

The repulsion technique can be used to alleviate this problem. The underlying idea is intuitively appealing and its implementation is straightforward: after the detection of a minimizer, in addition to the application of deflection or stretching, as described

TABLE I
PSEUDOCODE OF THE REPULSION PROCEDURE

Input: $\mathbb{X}^*, \mathbb{S}, r_{ij}$ and $p_{ij}, i = 1, \ldots, N$, and $j = 1, \ldots, m$

For $(i = 1 : N)$ **Do,**
  If $(\mathbb{X}^* \neq \emptyset)$ **Then**
    Compute $d_{ij} = \|X_i - X_j^*\|, j = 1, \ldots, m$
    For $(j = 1 : m)$ **Do,**
      If $(d_{ij} \leqslant r_{ij})$ **Then**
        Set $Z_{ij} = \frac{X_i - X_j^*}{d_{ij}}, j = 1, \ldots, m$
        Set $X_i = X_i + p_{ij} Z_{ij}$
      End If
    End For
  End If
End For

in the previous sections, the repulsion algorithm is activated. This stage of the proposed approach, ensures that if a particle moves toward one of the detected minimizers it will be repelled away from it.

Let $\mathbb{X}^* = \{X_j^*; j = 1, \ldots, m\}$ be the set of already detected minimizers, and $\mathbb{S} = \{X_i; i = 1, \ldots, N\}$ be the swarm, at a given iteration. The pseudocode of Table I is incorporated in the PSO algorithm, and executed after the determination of the new position of the swarm using (3) and (4), or, alternatively (5) and (6). The procedure is straightforward. If some minimizers have already been detected, then for the ith particle X_i, $i = 1, \ldots, N$, do: compute the norm $\|X_i - X_j^*\|$ for every minimizer $X_j^*, j = 1, \ldots, m$, to check if X_i lies in its *repulsion area*, whose range is determined by a prespecified parameter r_{ij}. If the particle lies within the bounds of the repulsion area, then it is repelled away from its center. This is achieved by adding the vector $p_{ij} Z_{ij}$ to X_i, where p_{ij} denotes a prespecified constant, which determines the strength of the repulsion, and Z_{ij} is the unitary vector with direction from X_j^* toward X_i. The vector Z_{ij} can be computed by dividing the vector $(X_i - X_j^*)$ with its norm. In practice, the values of r_{ij} and p_{ij}, are often fixed for all particles and detected minimizers. Clearly, information about the distribution of the minimizers in the search space can improve parameter configuration. If no information is available, then small values are preferred, especially for r_{ij}, to avoid enclosing minimizers not yet detected within the repulsion area of a detected minimizer. Moreover, p_{ij} shall be selected large enough to repel a particle away from the repulsion area of a minimizer. This will be illustrated experimentally in the next section.

The repulsion technique works in conjunction with the techniques presented in previous sections, and can be easily incorporated in the PSO algorithm. Moreover, using repulsion, the problem of applying deflection on a zero valued global minimizer does not affect the performance of the algorithm, since particles which move toward the detected minimizer will be repelled away from it.

IV. EXPERIMENTAL RESULTS

Problems arising in diverse scientific fields were selected to investigate the performance of the proposed approach. In all problems, the global minimum of the objective function was known *a priori*. The main goal was, thus, to detect several (or all) global minimizers, while alleviating local ones through the deflection and the stretching technique, in conjunction with the repulsion approach.

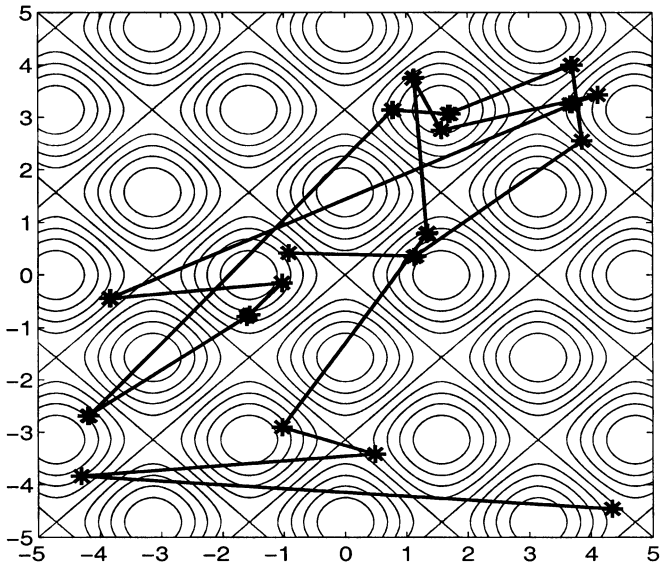

Fig. 17. Contour plot of the function defined in (16) and the rambling movement of a single particle of the swarm.

A. Computing Several Global Minimizers of Objective Functions

First, we have chosen a simple problem in order to better illustrate our approach. This problem has the characteristic that the objective function is identical in the neighborhoods of all global minimizers.

Let f be the function defined by [65]

$$f(X) = \cos^2(x_1) + \sin^2(x_2) \tag{16}$$

where $X = (x_1, x_2)^\top \in S = [-5, 5]^2$. This is a simple function whose contour plot is illustrated in Fig. 17. This function has only global minimizers, at points $(\kappa_1(\pi/2), \kappa_2\pi)$, where $\kappa_1 = \pm 1, \pm 2, \ldots$, and $\kappa_2 = 0, \pm 1, \pm 2, \ldots$. Overall, there are 12 global minimizers in the region $[-5, 5]^2$.

The plethora of equally attractive areas in the search space often results in a rambling movement of the swarm, and inability to converge toward a minimizer, also illustrated in Fig. 17 for a single particle [65]. Furthermore, there is no guarantee that all minimizers will be detected after a number of independent experiments, since no information concerning previously detected minimizers is made available to the swarm. These problems can be efficiently alleviated through the proposed approach. Specifically, whenever a global or local minimizer of the objective function is detected, the deflection or the stretching technique is applied, respectively. Moreover, the detected minimizer becomes a repeller, applying the procedure described in Section III-C.

For the test function defined in (16), the values for parameters of the PSO, which were used in the experiments performed, are reported in Table II. The selected values of w, c_1, c_2, and χ, are considered proper default values and they are widely used in the relevant literature. The $V_{\max}$ parameter was set equal to

TABLE II
PSO PARAMETER CONFIGURATION FOR THE TEST PROBLEM DEFINED IN (16)

Swarm's size	20
Accuracy	10^{-4}
c_1, c_2	2.05
w	$1 \rightarrow 0.1$
χ	0.729
$V_{\max}$	5
Maximum Iter.	2000

TABLE III
RESULTS FOR THE TEST PROBLEM DEFINED IN (16)

PSO variant	minimizers number	mean total iterations	mean iterations per minimizer
PSO-Co	12	315	26.2
PSO-In	12	2533	211.1

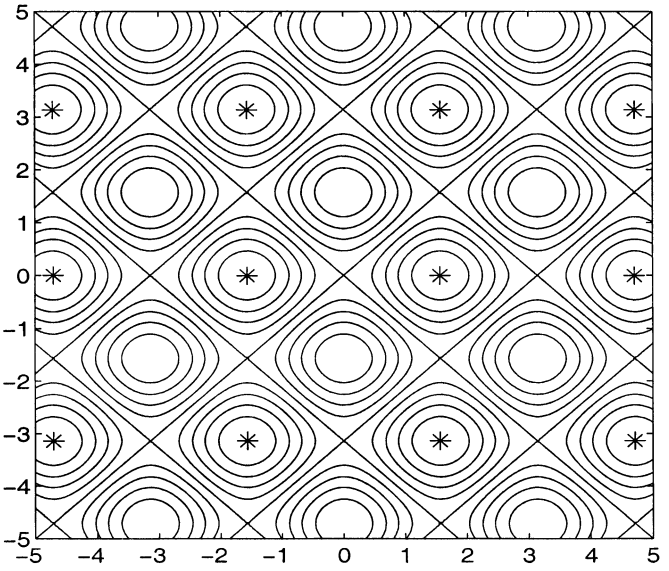

Fig. 18. Global minimizers (marked with asterisks) of the function defined in (16) that were obtained through the proposed approach.

half of the dynamic range of the particles. The desired accuracy was four decimal digits. The fixed parameters $r_{ij} = 0.5$ and $p_{ij} = 0.8$ were used for all particles. The number of detected minimizers, the total number of iterations required, as well as the mean number of the iterations required per minimizer for both the constriction factor and the inertia weight variants of the PSO (denoted as PSO-Co and PSO-In, respectively), are reported in Table III. All results are averaged over 30 independent experiments. The obtained global minimizers are illustrated in Fig. 18.

In all runs, all 12 global minimizers were obtained. Different values of r_{ij} and p_{ij} were also employed to investigate their effect on the algorithm's performance. For this purpose, a small value of r_{ij}, equal to 0.1, as well as a large value equal to 4 were used. The obtained results under these configurations were in line with the theoretically expected behavior of the algorithm which is described at the end of Section III-C. Specifically, the small value, $r_{ij} = 0.1$ could not prevent PSO from getting stuck at the local minima introduced by the deflection transformation. Thus, the algorithm was able to detect all 12 global minimizers only in 21 out of 30 experiments. On the other hand, taking $r_{ij} = 4$ resulted in enclosing global minimizers in the repulsion area of already detected minimizers and preventing PSO to converge to them. Consequently, the algorithm was able to detect all global minimizers in just 3 out of 30 experiments. These results support the claim that the optimal values of the repulsion parameters are problem dependent. In the specific test function, a choice of r_{ij} higher than the mean distance of the global minimizers per dimension (which is around 3), was responsible for the low performance of the algorithm. On the other hand, the small value also resulted in decreased performance, but its effect can be addressed by using additionally the stretching technique or applying deflection on the artificially introduced local minimizers. Consequently, if no information is available regarding the objective function, small values of r_{ij} are considered as a good initial choice.

Further experiments were performed using different values of the other repulsion parameter p_{ij}. Specifically, a very small value equal to 0.1, as well as higher values equal to 3 and 5 were used. The results verified that small values of p_{ij} may deter the particles from escaping from the repulsion area of a minimizer. Using $p_{ij} = 0.1$, PSO was able to detect all 12 minimizers in 22 out of 30 experiments, while using $p_{ij} = 3$ and $p_{ij} = 5$ resulted in the detection of all minimizers in every experiment.

The filled functions technique was also investigated in order to provide a benchmark for comparison with the proposed technique. The results produced were not very promising, since, for all experiments, the maximum number of detected minimizers never exceeded 4. Consequently, this approach was abandoned.

The proposed technique was further applied on the *Levy no. 5* function, defined in (15), using the aforementioned parameter setup. This function is considered hard due to the plethora of local minimizers (it has approximately 760 local minimizers and just one global in $[-10, 10]^2$). Using the proposed technique, both the constriction factor and inertia weight variant of PSO detected the global minimizer $x^* = (-1.307\,132, -1.424\,898)^\top$, in all experiments, avoiding all local minimizers, and especially a "bad" one at point $x = (-1.306\,853, 4.855\,487)^\top$, which is located nearby the global minimizer and has very low function value. The number of detected minimizers, the total number of iterations required, as well as the mean number of the iterations required per minimizer for both the constriction factor and the inertia weight variant of the PSO, for cases at which at least one local minimizer was detected and alleviated, are reported in Table IV.

Another interesting problem is the *Levy no. 3* function, which is defined as

$$f(x) = \left(\sum_{i=1}^{5} i\cos\left((i-1)x_1 + i\right)\right) \times \left(\sum_{i=1}^{5} i\cos\left((i+1)x_2 + i\right)\right). \tag{17}$$

This function has a multitude of global and local minimizers. The proposed technique was applied in order to obtain 10 global minimizers within $[-10, 10]^2$. The number of detected minimizers, the total number of iterations required, as well as the

TABLE IV
RESULTS FOR THE LEVY NO. 5 FUNCTION, DEFINED IN (15)

PSO variant	minimizers number	mean total iterations	mean iterations per minimizer
PSO-Co	2	3026.5	1036.5
PSO-In	2	3246.5	1246.5

TABLE V
RESULTS FOR THE LEVY NO. 3 FUNCTION, DEFINED IN (17)

PSO variant	minimizers number	mean total iterations	mean iterations per minimizer
PSO-Co	10	2913	291.3
PSO-In	10	6935	693.5

mean number of the iterations required per minimizer are reported in Table V.

B. Computing Periodic Orbits of Nonlinear Mappings

Nonlinear mappings are used to model conservative or dissipative dynamical systems [69]–[77]. Central role in the analysis of such mappings is played by points, which are invariant under the mapping, called *fixed points* or *periodic orbits*. A point $X = (x_1, \ldots, x_n)^\top$ is a fixed point of a mapping $\Phi(X)$, if $\Phi(X) = X$, and it is a periodic orbit of *period* p (or fixed point of *order* p), if

$$X = \Phi^p(X) = \underbrace{\Phi\left(\Phi(\ldots(\Phi(X))\ldots)\right)}_{p \text{ times}}.$$

Computing periodic orbits of nonlinear mappings is one of the most challenging problems of the nonlinear science, because analytic expressions for evaluating periodic orbits can be obtained only if the mapping is a polynomial of low degree and the period is low. Traditional methods, such as the Newton-family methods and related classes of algorithms, often fail, since they are affected by the mapping evaluations assuming large values in the neighborhood of saddle-hyperbolic periodic orbits, which are unstable in the linear approximation. Generally, the failure of these methods can also be attributed to the nonexistence of derivatives or poorly behaved partial derivatives in the neighborhood of the fixed point. The problem of finding more than one periodic orbit of a specific type (stable or unstable [78]) is very important, and cannot be tackled through the traditional algorithms or PSO without any additional technique.

Computing periodic orbits can be equivalently addressed as a global optimization problem. Indeed, if $\Phi = (\Phi_1, \ldots, \Phi_n)^\top : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear mapping, $X = (x_1, \ldots, x_n)^\top$ is a periodic orbit of period p, and $\Theta_n = (0, \ldots, 0)^\top$ is the origin, then it holds that

$$\Phi^p(X) = X \Rightarrow \Phi^p(X) - X = \Theta_n$$

or equivalently

$$\begin{pmatrix} \Phi_1^p(X) \\ \vdots \\ \Phi_n^p(X) \end{pmatrix} - \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{18}$$

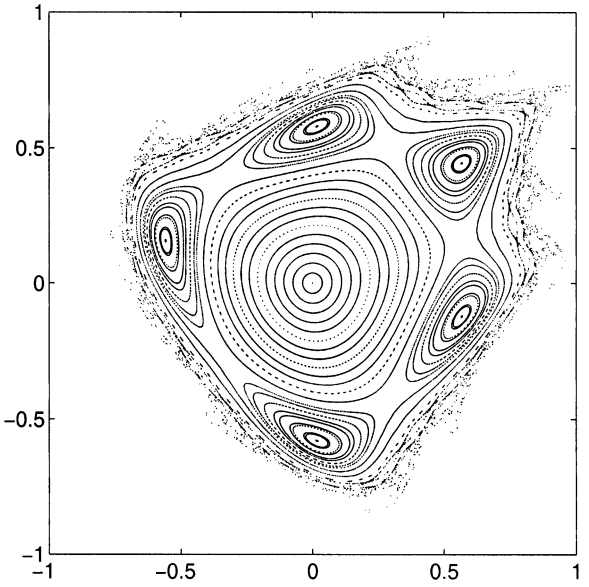

Fig. 19. Phase plot of the Hénon mapping for $\cos\alpha = 0.24$.

Thus, the problem of computing periodic orbits is equivalent to computing zeros of the nonlinear system

$$\begin{cases} \Phi_1^p(X) - x_1 = 0 \\ \quad\vdots \\ \Phi_n^p(X) - x_n = 0 \end{cases} \tag{19}$$

which, in turn, is equivalent to computing the global minimizers of the nonnegative function

$$f(X) = \sum_{i=1}^{n} \left(\Phi_i^p(X) - x_i\right)^2 = 0. \tag{20}$$

The global minimizers of the function f have zero function value, and the corresponding global minimizers are all periodic orbits of period p, of the nonlinear mapping Φ.

Consider the 2-D *Hénon* mapping, defined by

$$\Phi(X) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 - x_1^2 \end{pmatrix} \tag{21}$$

where α is the angle. This is an interesting model, which is related to the particle beams passing repeatedly through FODO cells of magnetic focusing elements [74]–[77].

The phase plot of the mapping for $\cos\alpha = 0.24$ is illustrated in Fig. 19, while the contour plot of the corresponding objective function, defined in (20), for period $p = 5$, is displayed in Fig. 20.

Another very interesting case is the 2-D Hénon mapping with $\cos\alpha = 0.8$. The phase plot of this mapping is illustrated in Fig. 21, while the corresponding objective function, for period $p = 1$, is illustrated in Fig. 22. This is a very difficult problem, because most algorithms tend to converge to the stable fixed point located at the origin or they are attracted by infinity, ignoring the unstable periodic orbit which lies in the very narrow channel at the right-hand side of the contour plot [74].

The aforementioned problems can be efficiently tackled through the proposed technique. If a periodic orbit of a specific type is computed, then all other periodic orbits of the same period and type can be obtained by applying the mapping

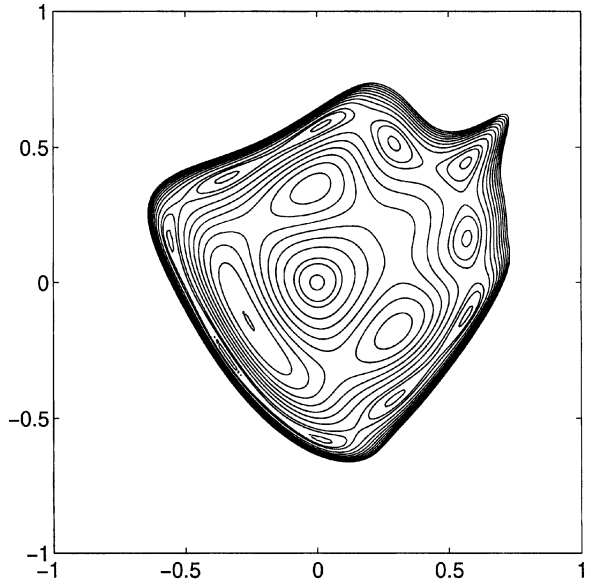

Fig. 20. Contour plot of the objective function for the Hénon mapping for period $p = 5$ and $\cos\alpha = 0.24$.

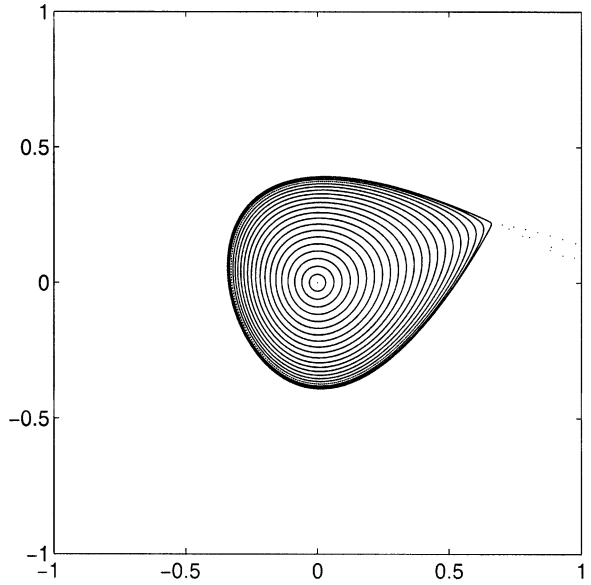

Fig. 21. Phase plot of the Hénon mapping for $\cos\alpha = 0.8$.

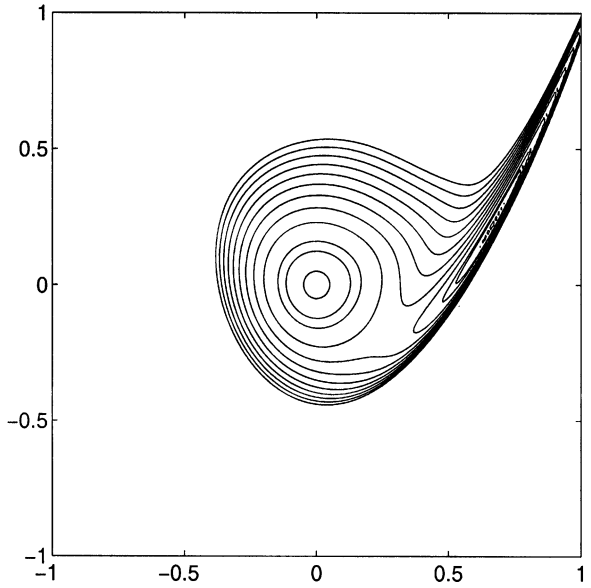

Fig. 22. Contour plot of the objective function for the Hénon mapping for period $p = 1$ and $\cos\alpha = 0.8$.

TABLE VI
CONFIGURATION OF THE PSO PARAMETERS FOR COMPUTING PERIODIC ORBITS OF PERIOD $p = 5$, OF THE HÉNON MAPPING, FOR $\cos\alpha = 0.24$

Swarm's size	20
Accuracy	10^{-4}
c_1, c_2	2.05
w	$1 \rightarrow 0.1$
χ	0.729
$V_{\max}$	1
Maximum Iter.	2000

TABLE VII
RESULTS FOR COMPUTING PERIODIC ORBITS OF PERIOD $p = 5$, OF THE HÉNON MAPPING, FOR $\cos\alpha = 0.24$

PSO variant	minimizers number	mean total iterations	mean iterations per minimizer
PSO-Co	11	249	22.6
PSO-In	11	1671	151.9

on the periodic orbit. If the proposed technique is applied on these points, then in the next run, convergence to any of these periodic orbits will be avoided, and hopefully an orbit of a different type will be computed.

The configuration of the PSO parameters for computing the periodic orbits of period $p = 5$, of the Hénon mapping for $\cos\alpha = 0.24$, are reported in Table VI. Once more, default values for the parameters were selected. In total, 11 global minimizers of the corresponding objective function were detected, five stable, five unstable, and a periodic orbit of period 1, located at the origin, which, of course, has also period 5 (multitude of p). For the repulsion technique, the values $p_{ij} = 0.08$ and $r_{ij} = 0.05$ were selected, since the search space is smaller in comparison to that of the previous test problem. The inclusion of enhanced techniques, like Topological Degree theory [79]–[81], may provide with a hint of the total number of global minimizers of the problem and, thus, render the selection of proper repulsion parameters easier. The total number of detected minimizers, the total number of iterations required, and the mean number of the iterations required per minimizer are reported in Table VII.

A second experiment, focused on the detection of periodic orbits of period $p = 1$ of the Hénon mapping for $\cos\alpha = 0.8$. The parameters' configuration was unaltered from the previous experiment. The proposed technique detected both periodic orbits. The total number of computed minimizers, the total number of iterations required, and the mean number of the iterations required per minimizer, are reported in Table VIII.

C. Computing Nash Equilibria in Finite Strategic Games

A finite *k-person strategic game* is defined by a set $\mathcal{K} = \{1, \ldots, k\}$ of *players*, each of whom has a *strategy set* $S_i = \{S_{i1}, \ldots, S_{im_i}\}$, consisting of m_i *pure strategies*

TABLE VIII
COMPUTING PERIODIC ORBITS OF PERIOD $p = 1$, OF THE HÉNON MAPPING, FOR $\cos\alpha = 0.8$

PSO variant	minimizers number	mean total iterations	mean iterations per minimizer
PSO-Co	2	21	10.5
PSO-In	2	88	44.0

TABLE IX
CONFIGURATION OF THE PSO PARAMETERS FOR COMPUTING NASH EQUILIBRIA OF THE "BATTLE OF SEXES" GAME

Swarm's size	20
Accuracy	10^{-4}
c_1, c_2	2.05
w	$1 \rightarrow 0.1$
χ	0.729
$V_{\max}$	0.5
Maximum Iter.	2000

[82]–[84]. Each player has a payoff function, $u_i : S \rightarrow \mathbb{R}$, where $S = S_1 \times \cdots \times S_k$. The payoff function can be extended to have domain $\mathbb{R}^m$, by the following relation:

$$u_i(p) = \sum_{s \in S} p(s) u_i(s)$$

where $p(s) = p_1(S_1) \times \cdots \times p_k(S_k)$, and p_i is a probability measure defined on S_i. In general, $p \in \mathcal{P}$, where $\mathcal{P} = \mathcal{P}_1 \times \cdots \times \mathcal{P}_k$, and $\mathcal{P}_i$ is the set of real valued functions on S_i. The set $\mathcal{P}$ is isomorphic to $\mathbb{R}^m$. Thus, we can write $p = (p_1, \ldots, p_k)$, where $p_i = (p_{i1}, \ldots, p_{im_i}) \in \mathcal{P}_i$. If $p \in \mathcal{P}$ and $p'_i \in \mathcal{P}_i$, then the notation (p'_i, p_{-i}) is used to denote the element $q \in \mathcal{P}$, satisfying $q_i = p'_i$, and $q_j = p_j$, for $j \neq i$.

A point $p^* \in \mathcal{P}$ is a *Nash equilibrium* of the game, if $p^* \in \Delta$, where $\Delta = \Delta_1 \times \cdots \times \Delta_k$, with $\Delta_i = \{p_i \in \mathcal{P}_i : \sum_j p_{ij} = 1, p_i \geqslant 0\}$, and for all $i \in \mathcal{K}$ and all $p_i \in \Delta_i$, the relation $u_i(p_i, p^*_{-i}) \leqslant u_i(p^*)$ holds [83].

The computation of Nash equilibria can be transformed in the problem of detecting global minimizers of an objective function. Specifically, for any $p \in \mathcal{P}$, $i \in \mathcal{K}$, and $s_{ij} \in S_i$, we can define the following functions:

$$x_{ij}(p) = u_i(s_{ij}, p_{-i}) \tag{22}$$
$$z_{ij}(p) = x_{ij}(p) - u_i(p) \tag{23}$$
$$g_{ij}(p) = \max\{z_{ij}(p), 0\}. \tag{24}$$

Then, a Nash equilibrium is a global minimizer of the objective function $v : \Delta \rightarrow \mathbb{R}$, which is defined by [83]

$$v(p) = \sum_{i \in \mathcal{K}} \sum_{1 \leqslant j \leqslant m_i} g^2_{ij}(p). \tag{25}$$

The "Battle of Sexes" is a game of two players with two strategies per player. The payoff matrix for each player is

$$U_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad U_2 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

This game has three Nash equilibria. Two of which are pure strategy equilibria and one is a mixed strategy equilibium. Thus, the corresponding four-dimensional optimization problem has three global minimizers.

TABLE X
RESULTS FOR COMPUTING NASH EQUILIBRIA OF THE "BATTLE OF SEXES" GAME

PSO variant	minimizers number	mean total iterations	mean iterations per minimizer
PSO-Co	3	52	17.3
PSO-In	3	707	235.6

TABLE XI
RESULTS FOR COMPUTING NASH EQUILIBRIA OF THE GAME WITH THREE PLAYERS AND TWO STRATEGIES FOR EACH OF THEM

PSO variant	minimizers number	mean total iterations	mean iterations per minimizer
PSO-Co	3	204	68.0
PSO-In	3	64	21.33

The configuration of PSO parameters for the "Battle of Sexes" game is reported in Table IX and the results are reported in Table X. Although two of the global minimizers are exactly at the boundary of the search space, the proposed technique was able to detect all three Nash equilibria efficiently.

The proposed approach was also applied on a different game of three players with two pure strategies each. In this case, the payoff matrices are three-dimensional $(2 \times 2 \times 2)$ and they are defined as

$$U_1^1 = \begin{pmatrix} 1 & 0.5 \\ 3 & 1 \end{pmatrix}, \quad U_1^2 = \begin{pmatrix} 5 & 0 \\ 4 & 2 \end{pmatrix},$$
$$U_2^1 = \begin{pmatrix} 0 & 0.5 \\ 1 & 1 \end{pmatrix}, \quad U_2^2 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix},$$
$$U_3^1 = \begin{pmatrix} 1 & 0.3 \\ 5 & 1 \end{pmatrix}, \quad U_3^2 = \begin{pmatrix} 0 & 1 \\ 3 & 4 \end{pmatrix}.$$

The corresponding objective function is six-dimensional. The proposed approach was applied to compute three Nash equilibria. The results are reported in Table XI. An interesting point worth noting is that the proposed approach managed to detect Nash equilibria that could not be detected by other established techniques, which are based on deterministic algorithms.

V. CONCLUSION

In this paper, an approach aiming at the computation of all global minimizers, while avoiding local minimizers, in the context of the PSO algorithm, is proposed and its performance on several problems originating from diverse scientific fields is investigated. The approach incorporates the recently proposed *deflection* and *stretching* techniques to overcome local minimizers, as well as a *repulsion* technique to repel particles away from previously detected minimizers. This approach contributes to an expanding area of research of intense interdisciplinary scientific interest. Experimental results indicate that this is an effective approach for computing more than one global minimizers that can be used to solve high-edge problems of nonlinear science and economic theory.

Incorporation of Topological Degree theory [79]–[81] to obtain information on the number, as well as the location of the

minimizers will be considered in future work to compute with certainty all global minimizers of a function.

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers for their helpful remarks and comments. They also thank Dr. G. S. Androulakis, Dr. G. D. Magoulas, and Dr. V. P. Plagianakos for useful discussions.

REFERENCES

[1] R. Horst and H. Tuy, *Global Optimization—Deterministic Approaches*. New York: Springer-Verlag, 1996.
[2] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. New York: Oxford Univ. Press, 1996.
[3] H.-G. Beyer, *The Theory of Evolution Strategies*. Berlin, Germany: Springer-Verlag, 2001.
[4] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies: A comprehensive introduction," *Nat. Comput.*, vol. 1, no. 1, pp. 3–52, 2002.
[5] I. Rechenberg, "Evolution strategy," in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks II, and C. Robinson, Eds. Piscataway, NJ: IEEE Press, 1994, pp. 147–159.
[6] H.-P. Schwefel, *Numerical Optimization of Computer Models*. New York: Wiley, 1981.
[7] ——, *Evolution and Optimum Seeking*. New York: Wiley, 1995.
[8] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[9] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1994.
[10] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming—An Introduction*. San Mateo, CA: Morgan Kaufman, 1998.
[11] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.
[12] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press, 1996.
[13] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Symp. Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.
[14] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. 2002 IEEE Congr. Evolutionary Computation*, Honolulu, HI, 2002, pp. 1051–1056.
[15] J. E. Fieldsend and S. Singh, "A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence," in *Proc. 2002 U.K. Workshop on Computational Intelligence*, Birmingham, U.K., 2002, pp. 34–44.
[16] X. Hu, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proc. 2002 IEEE Congr. Evolutionary Computation*, Honolulu, HI, 2002.
[17] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method in multiobjective problems," in *Proc. 2002 ACM Symp. Applied Computing (SAC 2002)*, Madrid, Spain, 2002, pp. 603–607.
[18] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proc. IEEE 2003 Swarm Intelligence Symp.*, 2003, pp. 26–33.
[19] X. Li, "A nondominated sorting particle swarm optimizer for multiobjective optimization," in *Lecture Notes in Computer Science (LNCS)*, E. Cantú-Paz, J. A. Foster, K. Deb, D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. Dowsland, N. Jonoska, and J. Miller, Eds. New York: Springer-Verlag, 2003, vol. 2723, pp. 37–48.
[20] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimization for minimax problems," in *Proc. IEEE 2002 Congr. Evolutionary Computation* Honolulu , HI, 2002, pp. 1582–1587.
[21] Y. Shi and R. A. Krohling, "Co-evolutionary particle swarm optimization to solve min-max problems," in *Proc. 2002 IEEE Conf. Evolutionary Computation*, Honolulu, HI, May 2002, pp. 1682–1687.
[22] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimization for integer programming," in *Proc. IEEE 2002 Congr. Evolutionary Computation* Honolulu, HI, 2002, pp. 1576–1581.
[23] A. Carlisle, "Applying the particle swarm optimizer to non-stationary environments," Ph.D. dissertation, Auburn Univ., Auburn, AL, 2002.
[24] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proc. 2001 IEEE Congr. Evolutionary Computation* Seoul, South Korea, 2001, pp. 94–100.
[25] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimizer in noisy and continuously changing environments," in *Artificial Intelligence and Soft Computing*, M. Hamza, Ed. Cancun, Mexico: IASTED/ACTA Press, 2001, pp. 289–294.
[26] ——, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Comput.*, vol. 1, no. 2–3, pp. 235–306, 2002.
[27] K. E. Parsopoulos, E. C. Laskari, and M. N. Vrahatis, "Solving ℓ_1 norm errors-in-variables problems using particle swarm optimizer," in *Artificial Intelligence and Applications*, M. Hamza, Ed. Marbella, Spain: IASTED/ACTA Press, 2001, pp. 185–190.
[28] K. E. Parsopoulos and M. N. Vrahatis, "Investigating the existence of function roots using particle swarm optimization," in *Proc. IEEE 2003 Congr. Evolutionary Computation* Canberra, Australia, 2003, pp. 1448–1455.
[29] M. A. Abido, "Optimal design of power system stabilizers using particle swarm optimization," *IEEE Trans. Energy Conversion*, vol. 17, pp. 406–413, Sept. 2002.
[30] D. K. Agrafiotis and W. Cedeno, "Feature selection for structure-activity correlation using binary particle swarms," *J. Medicinal Chem.*, vol. 45, no. 5, pp. 1098–1107, 2002.
[31] A. R. Cockshott and B. E. Hartman, "Improving the fermentation medium for Echinocandin B production part II: Particle swarm optimization," *Process Biochem.*, vol. 36, pp. 661–669, 2001.
[32] P. C. Fourie and A. A. Groenwold, "The particle swarm optimization algorithm in size and shape optimization," *Struct. Multidisc. Optim.*, vol. 23, pp. 259–267, 2002.
[33] W. Z. Lu, H. Y. Fan, A. Y. T. Leung, and J. C. K. Wong, "Analysis of pollutant levels in central Hong Kong applying neural network method with particle swarm optimization," *Environ. Monitoring Assessment*, vol. 79, pp. 217–230, 2002.
[34] C. O. Ourique, E. C. Biscaia, and J. C. Pinto, "The use of particle swarm optimization for dynamical analysis in chemical processes," *Comput. Chem. Eng.*, vol. 26, pp. 1783–1793, 2002.
[35] K. E. Parsopoulos, E. I. Papageorgiou, P. P. Groumpos, and M. N. Vrahatis, "A first study of fuzzy cognitive maps learning using particle swarm optimization," in *Proc. IEEE 2003 Congr. Evolutionary Computation* Canberra, Australia, 2003, pp. 1440–1447.
[36] T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Eng. Opt.*, vol. 34, no. 2, pp. 141–153, 2002.
[37] A. Saldam, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocess. Microsyst.*, vol. 26, pp. 363–371, 2002.
[38] V. Tandon, H. El-Mounayri, and H. Kishawy, "NC end milling optimization using evolutionary computation," *Int. J. Mach. Tools Manuf.*, vol. 42, pp. 595–605, 2002.
[39] J. C. Tillett, R. M. Rao, F. Sahin, and T. M. Rao, "Particle swarm optimization for the clustering of wireless sensors," in *Proc. SPIE*, vol. 5100, Orlando, FL, 2003.
[40] S. E.S. Easter Selvan, S. Subramanian, and S. T.S. Theban Solomon, "Novel technique for PID tuning by particle swarm optimization," in *Proc. 7th Annu. Swarm Users/Researchers Conf. (SwarmFest 2003)*, Notre Dame, IN, 2003.
[41] R. C. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools*. New York: Academic, 1996.
[42] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks* Perth, Australia, 1995, vol. IV, pp. 1942–1948.
[43] ——, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.
[44] P. J. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences," in *Evolutionary Programming*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. VII, pp. 601–610.
[45] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Evolutionary Programming*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. VII, pp. 611–616.
[46] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. VII, pp. 591–600.
[47] ——, "A modified particle swarm optimizer," in *Proc. IEEE Conf. Evolutionary Computation* Anchorage, AK, 1998, pp. 69–73.
[48] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 58–73, Feb. 2002.
[49] J. Kennedy, "The behavior of particles," in *Evolutionary Programming*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. VII, pp. 581–590.

[50] K. E. Parsopoulos and M. N. Vrahatis, "Initializing the particle swarm optimizer using the nonlinear simplex method," in *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, A. Grmela and N. Mastorakis, Eds. Interlaken, Switzerland: WSEAS Press, 2002, pp. 216–221.
[51] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[52] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Inform. Process. Lett.*, vol. 85, pp. 317–325, 2003.
[53] Y. Shang and B. W. Wah, "Global optimization for neural network training," *IEEE Comput.*, vol. 29, no. 3, pp. 45–54, 1996.
[54] A. Törn and A. Žilinskas, *Global Optimization*. Berlin, Germany: Springer-Verlag, 1989.
[55] A. A. Goldstein and J. F. Price, "On descent from local minima," *Math. Comput.*, vol. 25, no. 3, pp. 569–574, 1971.
[56] L. B. Shusterman, "The method of successive elimination in search for the global optimum of multiextremal algebraic functions," *Radioelektronika*, vol. 79, no. 6, pp. 58–63, 1979.
[57] A. V. Vilkov, N. P. Zhidkov, and B. M. Shchedrin, "A method of search for the global minimum of a function of one variable," *J. Comp. Math. Math. Phys.*, vol. 75, pp. 1040–1043, 1975.
[58] S. Gomez and A. V. Levy, "The tunneling method for solving the constrained global optimization problem with several nonconnected feasible regions," in *Lecture Notes in Mathematics 909*. Berlin, Germany: Springer-Verlag, 1982, pp. 34–47.
[59] A. Montalvo, "Development of a new algorithm for the global minimization of functions," Ph.D. dissertation, Universidad Nacional Autonoma de Mexico, Mexico, 1979.
[60] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[61] R. P. Ge, "A filled function method for finding a global minimizer of a function of several variables," in *Math. Programming*, vol. 46, 1990, pp. 191–204.
[62] ——, "The theory of the filled function method for finding a global minimizer of a nonlinearly constrained minimization problem," *J. Comp. Math.*, vol. 5, pp. 1–9, 1987.
[63] R. P. Ge and Y. F. Qin, "A class of filled functions for finding global minimizers of a function of several variables," *JOTA*, vol. 54, pp. 241–252, 1987.
[64] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Objective function "stretching" to alleviate convergence to local minima," *Nonlinear Anal. Theory Meth. Appl.*, vol. 47, no. 5, pp. 3419–3424, 2001.
[65] K. E. Parsopoulos and M. N. Vrahatis, "Modification of the particle swarm optimizer for locating all the global minima," in *Artificial Neural Networks and Genetic Algorithms*, V. Kurkova, N. Steele, R. Neruda, and M. Karny, Eds. Wien, Germany: Springer-Verlag, 2001, pp. 324–327.
[66] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, "On the alleviation of local minima in backpropagation," *Nonlinear Anal. Theory Meth. Appl.*, vol. 30, no. 7, pp. 4545–4550, 1997.
[67] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Improving particle swarm optimizer by function "stretching"," in *Advances in Convex Analysis and Global Optimization*. ser. Nonconvex Optimization and Applications, N. Hadjisavvas and P. Pardalos, Eds. Norwell, MA: Kluwer-Academic, 2001, vol. 54, ch. 28, pp. 445–457.
[68] ——, "Stretching technique for obtaining global minimizers through particle swarm optimization," in *Proc. Particle Swarm Optimization Workshop*, Indianapolis, IN, 2001, pp. 22–29.
[69] G. D. Birkhoff, "Dynamical systems with two degrees of freedom," *Trans. Amer. Math. Soc.*, vol. 18, pp. 199–300, 1917.
[70] T. C. Bountis and R. H. Helleman, "On the stability of periodic orbits of two-dimensional mappings," *J. Math. Phys.*, vol. 22, no. 9, p. 1867, 1981.
[71] J. M. Greene, "A method for determining a stochastic transition," *J. Math. Phys.*, vol. 20, pp. 1183–1201, 1979.
[72] M. Hénon, "Numerical study of quadratic area-preserving mappings," *Quart. Appl. Math.*, vol. 27, pp. 291–311, 1969.
[73] K. E. Parsopoulos and M. N. Vrahatis, "Computing periodic orbits of nondifferentiable/discontinuous mappings through particle swarm optimization," in *Proc. IEEE Swarm Intelligence Symp.*, Indianapolis, IN, 2003, pp. 34–41.
[74] M. N. Vrahatis, "An efficient method for locating and computing periodic orbits of nonlinear mappings," *J. Comp. Phys.*, vol. 119, pp. 105–119, 1995.
[75] M. N. Vrahatis, T. C. Bountis, and M. Kollmann, "Periodic orbits and invariant surfaces of 4-D nonlinear mappings," *Int. J. Bifurc. Chaos*, vol. 6, pp. 1425–1437, 1996.
[76] M. N. Vrahatis, H. Isliker, and T. C. Bountis, "Structure and breakdown of invariant tori in a 4-D mapping model of accelerator dynamics," *Inter. J. Bifurc. Chaos*, vol. 7, pp. 2707–2722, 1997.
[77] M. N. Vrahatis, G. Servizi, G. Turchetti, and T. C. Bountis, "A procedure to compute the fixed points and visualize the orbits of a 2-D map," Eur. Org. Nucl. Res., CERN, Tech. Rep. SL/93-06 (AP), 1993.
[78] C. Skokos, "On the stability of periodic orbits of high dimensional autonomous hamiltonian systems," *Physica D*, vol. 159, no. 3–4, pp. 155–179, 2001.
[79] N. G. Lloyd, *Degree Theory*. Cambridge, U.K.: Cambridge Univ. Press, 1978.
[80] B. Mourrain, M. N. Vrahatis, and J. C. Yakoubsohn, "On the complexity of isolating real roots and computing with certainty the topological degree," *J. Complexity*, vol. 18, pp. 612–640, 2002.
[81] J. M. Ortega and W. C. Rheinbolt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic, 1970.
[82] R. D. Luce and H. Raiffa, *Games and Decisions*. New York: Dover, 1985.
[83] R. D. McKelvey and A. McLennan, "Computation of equilibria in finite games," in *Handbook of Computational Economics*. ser. Handbooks in Economics (13), H. M. Amman, D. A. Kendrick, and J. Rust, Eds. Amsterdam, The Netherlands: North-Holland, 1996, vol. 1, pp. 87–142.
[84] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA: MIT Press, 1994.

Konstantinos E. Parsopoulos received the Diploma degree in mathematics and the M.Sc. degree in mathematics of computers and decision making from the University of Patras, Patras, Greece, in 1998 and 2001, respectively, where he is currently working toward the Ph.D. degree.

He is a Member of the University of Patras Artificial Intelligence Research Center (UPAIRC). He was a Visiting Research Fellow at the Collaborative Research Center "Computational Intelligence" (SFB 531), Department of Computer Science, University of Dortmund, Dortmund, Germany, in 2001, and at the Institut National de Recherche en Informatique et en Automatique (INRIA), France, in 2003. His research interests include computational and swarm intelligence algorithms for optimization and applications.

Michael N. Vrahatis received the Diploma and Ph.D. degrees in mathematics from the University of Patras, Patras, Greece, in 1978 and 1982, respectively.

He has been a Professor with the Department of Mathematics, University of Patras, since August 2000. He was a Visiting Research Fellow with the Department of Mathematics, Cornell University, Ithaca, NY, from 1987 to 1988, and a Visiting Professor with the Istituto Nazionale di Fisica Nucleare (INFN), Bologna, Italy, in 1992, 1994, and 1998, respectively; the Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, in 1999, the Department of Ocean Engineering, Design Laboratory, Massachusetts Institute of Technology, Cambridge, in 2000, and the Collaborative Research Center "Computational Intelligence" (SFB 531), Department of Computer Science, University of Dortmund, Dortmund, Germany, in 2001. He was a Visiting Researcher at CERN (European Organization of Nuclear Research), Geneva, Switzerland, in 1992, and at the Institut National de Recherche en Informatique et en Automatique (INRIA), France, in 1998 and 2003. He is the author or (coauthor) of more than 200 publications in both pure and applied mathematics, including topological degree theory and its applications, numerical methods and software development, numerical analysis, numerical solution of systems of nonlinear algebraic, transcendental or differential equations and their applications on dynamical systems, accelerator dynamics and chaos, optimization, neural network training, genetic and evolutionary algorithms, data mining, cryptography, and artificial intelligence. He is among the founders of the University of Patras Artificial Intelligence Research Center (UPAIRC), established in 1997, where he currently serves as Director. He has been principal investigator of several research grants from the European Union and the Hellenic Ministry of Industry, Energy, and Technology.