Nonlinear Phenomena in Complex Systems, vol. **11**, no. 2 (2008), pp. 285 - 291

Stochastic Optimization for Detecting Periodic Orbits of Nonlinear Mappings

Y. G. Petalas,* K. E. Parsopoulos,† and M. N. Vrahatis‡
Computational Intelligence Laboratory (CI Lab), Department of Mathematics,
University of Patras, GR–26100 Patras, GREECE
(Received 30 January, 2008)

The detection of periodic orbits bears significance for the study of nonlinear mappings, since they can reveal crucial information on their dynamics. Recently, population–based stochastic optimization algorithms were introduced to address problems where traditional gradient–based approaches failed. The efficiency of these approaches in applications, triggered further research towards the development of more efficient variants. This work presents the principal concepts of applying concurrent stochastic population–based approaches for the detection of periodic orbits, and also reports new results attained by the application of Memetic Algorithms on well–known chaotic maps for periodic orbits with high period.

PACS numbers: 02.60.Pn, 05.45.Pq, 45.05.+x, 45.10.-b
Keywords: nonlinear mapping, periodic orbits, stochastic optimization

1. Introduction

The wide application of dynamical systems for the description and modeling of systems in different scientific disciplines has boosted the relative research significantly during the past years. In this framework, nonlinear mappings have been used extensively to model conservative or dissipative dynamical systems [1–10]. A crucial role in the study of such mappings is played by their periodic orbits, since they can provide useful information regarding the geometric and dynamical properties of the mapping, especially in cases of chaotic behavior [11].

The detection of periodic orbits is a very challenging task, since analytic expressions are available only in a limited number of cases that refer mostly to polynomial mappings of low degree. For this purpose, traditional root finding algorithms, such as the Newton–family methods, have been widely applied for computing periodic orbits numerically. However, such algorithms fail in cases where the mapping lacks of nice mathematical characteristics, such as continuity and differentiability.

Recently, population–based stochastic optimization algorithms have been introduced as an alternative tool for computing periodic orbits of nonlinear mappings, through the transformation of the corresponding root finding problem to an equivalent global minimization problem [12, 13]. The requirement of function values solely, along with their ability to perform better global search than traditional gradient–based approaches, renders these algorithms a very appealing alternative. Applications on well–known mappings, as well as on galactic potentials, justified their wide applicability and efficiency [12–14].

Population–based algorithms can also be equipped with a local search component that enhances their search ability, especially when further fine–tuning of the already detected solutions is required in order to achieve higher accuracy of the solutions. This class of algorithms is named *Memetic Algorithms* (MAs) and it has been shown to be very efficient in different application fields [14–16]. Usually, MAs consist of a global component that performs a rough search of the search space and a local search algorithm that is evoked occasionally to refine the already detected promising solutions. This effect can be very useful when high accuracy of the solutions is required.

In this paper, we present the established framework for detecting periodic orbits using population–based stochastic optimization algorithms. We also demonstrate the application of two popular algorithms, namely Particle Swarm Optimization (PSO) and Differential Evolution (DE), along with their memetic counterparts, for detecting long periodic orbits of well–known mappings. In the memetic approaches, the Solis and Wets algorithm is used as the local search component, along with an entropy–based variant that has been shown to be very efficient in different applications [14].

The rest of the paper is organized as follows:

*E-mail: `petalas@math.upatras.gr`
†E-mail: `kostasp@math.upatras.gr`
‡E-mail: `vrahatis@math.upatras.gr`; Also at University of Patras Artificial Intelligence Research Center (UPAIRC)

the general framework of detecting periodic orbits of nonlinear mappings through optimization algorithms is presented in Section 2, while the considered algorithms as well as their memetic variants are roughly described in Section 3. Experimental results on well–known mappings are reported in Section 4, and the paper concludes in Section 5.

2. The established optimization framework

Let $\Phi(X)$ be an n–dimensional mapping,

$$\Phi(X) = \Big(\Phi_1(X), \ldots, \Phi_n(X)\Big)^\top : \mathbb{R}^n \to \mathbb{R}^n.$$

Then, a point $X = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$, is called a *fixed point of order* p or *periodic orbit of period* p of $\Phi(X)$, if

$$X = \Phi^p(X) \equiv \underbrace{\Phi(\Phi(\ldots\Phi(\Phi(X))\ldots))}_{p \text{ times}}, \qquad (1)$$

i.e., p consecutive applications of the mapping on the point X, produce the same initial point X.

Let $\Theta_n = (0, \ldots, 0)^\top$ be the origin of $\mathbb{R}^n$. Then, Eq. (1) implies that [1]

$$\Phi^p(X) - X = \Theta_n \Rightarrow \begin{cases} \Phi_1^p(X) - x_1 = 0, \\ \qquad \vdots \\ \Phi_n^p(X) - x_n = 0. \end{cases} \qquad (2)$$

Therefore, detecting a periodic orbit of period p of the mapping $\Phi(X)$, is equivalent to finding the solutions of the nonlinear system defined by Eq. (2).

The aforementioned problem can be transformed to an equivalent global minimization problem, by applying the following transformation [17]:

$$f(X) = \sum_{i=1}^{n} \Big(\Phi_i^p(X) - x_i\Big)^2. \qquad (3)$$

The produced function $f(X)$ is non-negative and all its global minimizers, for which $f(X) = 0$, are periodic orbits of period p of the mapping Φ. Thus, minimizing f is equivalent to computing the periodic orbits of $\Phi(X)$.

An advantage of bringing the problem into an optimization framework is the exploitation of the existence of chaos to the benefit of the employed optimization algorithm. This is due to the nature of the objective function defined by Eq. (3), which is steeper in the neighborhood of periodic orbits within chaotic regions, since significantly high function values are assumed by points very close to the periodic orbit, producing steep, narrow valleys around it that can help the optimizer to converge rapidly towards the periodic orbit. Moreover, the transformation to an optimization problem permits the application of a wide variety of both deterministic and stochastic optimization algorithms. Also, the application of techniques such as Deflection and Stretching [17] that can sequentially detect periodic orbits of the same period but different stability features is possible through the transformation to an optimization problem.

Although a differentiable norm was used for the derivation of Eq. (3) in [17], this is not obligatory since stochastic optimization algorithms can solve the optimization problem efficiently, requiring only the value of the objective function $f(X)$, which can be even discontinuous. Thus, any ℓ_q–norm, for a real number $q \geqslant 1$, which is defined as

$$\begin{aligned} \|\Phi^p(X)\|_q &= \left(\sum_{i=1}^{n} |\Phi_i^p(X) - X_i|^q\right)^{1/q}, \\ \|\Phi^p(X)\|_\infty &= \max_{1 \leqslant i \leqslant n} |\Phi_i^p(X) - X_i|, \end{aligned}$$

can be used instead of the squared ℓ_2–norm of Eq. (3). Recently, such approaches that use population–based algorithms were proposed. Most works considered concurrent population–based stochastic optimization algorithms, instead of the traditional single–point point stochastic approaches (such as Simulated Annealing) [12, 13, 18]. This choice was based on their ability to evolve concurrently many search points, performing better exploration, while they can be parallelized easily, exhibiting remarkable efficiency in applications where the mapping evaluation is time–consuming (e.g., multiple integrations are required for computing the mapping) [12].

Such promising algorithms that have already been used to tackle the optimization problem successfully, along with their memetic variants that will be illustrated in the experiments, are described in the following sections.

3. The considered algorithms

In this section we describe the Particle Swarm Optimization and the Differential Evolution algorithm, along with their memetic counterparts that incorporate the Solis and Wets local search algorithm. The workings of these algorithms will be illustrated later for the detection of long periodic orbits of well–known nonlinear mappings.

3.1. Particle swarm optimization

PSO is a stochastic population–based optimization algorithm that exploits a population of individuals to explore the search space [19]. Its development was initially inspired by the flocking behavior of living organisms as well as particle physics, therefore it is categorized as a *swarm intelligence* algorithm [20]. In PSO's framework, the population is called a *swarm* and the individuals (i.e., the search points) are called *particles*.

Each particle moves with an adaptable velocity (position shift) within the search space, while it stores into a memory the best position it has ever visited. Also, it exchanges information with the rest of the particles (or just a portion of the swarm). This group of communicating particles are called its *neighborhood* and the communicated information is the best position ever achieved by the whole neighborhood. In the optimization framework, such good positions are characterized by lower function values. The case where the whole swarm is considered as the neighborhood for each particle is called the *global* PSO variant, while smaller neighborhoods define *local* PSO variants.

The neighborhoods are usually defined in the space of particles' indices rather than in actual search space, based on a user–defined *neighborhood topology*. The reason for this selection is the alleviation of the heavy computational burden required for the computation of all distances among particles, as well as the promotion of diversity in the swarm. The most common neighborhood topology is the "ring topology", where the particles are assumed to lie on a ring, having two immediate neighbors each. This topology was also adopted in our experiments.

Let $\mathcal{S} \subset \mathbb{R}^n$ be the search space, $f : \mathcal{S} \to \mathbb{R}$ be the objective function, and $\mathbb{S}$ be a swarm consisting of M particles, $\mathbb{S} = \{X_1, \ldots, X_M\}$, with $X_i \in \mathcal{S}$, $i = 1, 2, \ldots, M$. The velocity of this particle is also an n–dimensional vector, V_i, and the best previous position encountered by the i–th particle in $\mathcal{S}$ is denoted by $P_i \in \mathcal{S}$. Assume g_i to be the index of the particle that attained the best previous position among all the particles in the neighborhood of X_i, i.e., $f(P_{g_i}(t)) \leqslant f(P_j(t))$, for all neighbors X_j of X_i, and t to be the iteration counter. Then, the swarm is manipulated by the equations [21]:

$$V_i(t+1) = \chi \Big[V_i(t) + c_1 r_1 \Big(P_i(t) - X_i(t) \Big) + c_2 r_2 \Big(P_{g_i}(t) - X_i(t) \Big) \Big], \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \quad (5)$$

where $i = 1, 2, \ldots, M$; χ is a parameter called *constriction factor*; c_1 and c_2 are two positive constants called *cognitive* and *social* parameter, respectively; and r_1, r_2, are random numbers drawn from a uniform distribution in $[0, 1]$. All vector operations are performed componentwise. The constriction factor is a mechanism for controlling the magnitude of the velocities. Stability analysis of PSO provided different configurations of χ with respect to the parameters c_1 and c_2 [21].

In the global variant of PSO, all particles are attracted by the same overall best position, converging faster towards specific points. Thus, the global variant of PSO emphasizes exploitation over exploration. On the other hand, in the local variant, the information of the best position of each neighborhood is transmitted to the other particles of the swarm through their neighbors. Therefore, the attraction to specific best positions is weaker, hindering the swarm from getting stuck in locally optimal solutions. Thus, the local variant of PSO emphasizes exploration over exploitation. The balance between these two properties is crucial for the performance of each global optimization algorithm [17, 22, 23].

3.2. The differential evolution algorithm

The DE algorithm was developed by Storn and Price [24]. It is a direct numerical search method, which utilizes M, n–dimensional parameter vectors X_i, $i = 1, 2, \ldots, M$, to probe the search space. The initial population is taken to be uniformly distributed in the search space. At each iteration (generation) of the algorithm, two operators, namely *mutation* and *recombination*, are applied on the individuals, and a new population arises. Then, *selection* takes place, and the best M individuals from the old and the new populations are selected to comprise the next generation.

According to the *mutation* operator, for each vector $X_i(t)$, $i = 1, 2, \ldots, M$, at iteration t, a *mutant vector* is produced by

$$V_i(t+1) = X_{r_1}(t) + K \Big(X_{r_2}(t) - X_{r_3}(t) \Big),$$

where $r_1, r_2, r_3 \in \{1, 2, \ldots, M\}$ are mutually different random indices and $K \in (0, 2]$. The indices r_1, r_2, r_3, also need to differ from the current index, i, and, consequently, mutation can be applied only if $M \geqslant 4$.

Following the mutation phase, the *recombination* operator is applied on the population. Thus, a *trial vector*,

$$U_i(t+1) = \Big(U_{i1}(t+1), U_{i2}(t+1), \ldots, U_{in}(t+1) \Big),$$

is generated, where

$$U_{ij}(t+1) = \begin{cases} V_{ij}(t+1), & \text{if } R_j \leqslant CR \text{ or } j = \mathrm{RI}(i), \\ X_{ij}(t), & \text{if } R_j > CR \text{ and } j \neq \mathrm{RI}(i), \end{cases}$$

where $j = 1, 2, \ldots, n$; R_j is the j–th evaluation of a uniform random number generator in the range $[0, 1]$; CR is the (user specified) crossover constant within $[0, 1]$; and $\mathrm{RI}(i)$ is a randomly chosen index within $\{1, 2, \ldots, n\}$.

To decide whether or not the vector $U_i(t+1)$ should be a member of the population comprising the next generation, it is compared to the initial vector $X_i(t)$, and

$$X_i(t+1) = \begin{cases} U_i(t+1), & \text{if } f(U_i(t+1)) \\ & \quad < f(X_i(t)), \\ X_i(t), & \text{otherwise,} \end{cases}$$

where f is the objective function under consideration.

The procedure described above is considered as the standard variant of the DE algorithm, denoted as DE/rand/1/bin, and it was used in our experiments. Different kinds of mutation and crossover operators have also been applied on minimization problems with promising results [24].

Table 1. Results of PSO for TP1

Norm	SS	Max	Min	Mean	StD	Suc
ℓ_2	40	83680	1760	11827.1	16968.8	93
	60	78360	3780	11070.0	11929.7	100
ℓ_1	40	94680	2560	9222.7	13852.9	97
	60	64740	3240	9003.1	7057.1	98
ℓ_∞	40	65880	2440	7279.2	7669.1	97
	60	66300	4440	9118.2	8140.7	100
$M\ell_\infty$	40	9408	598	2774.9	1602.8	100

3.3. Memetic algorithms

MAs draw inspiration from natural adaptation models that combine individual evolution and learning within a lifetime [25]. They consist of a global and a local search component. The first emulates evolution of the individual, while personal refinement is encouraged through the latter [25]. Thus, MAs strive to balance the trade–off between exploration and exploitation, efficiently. For this purpose, evolutionary algorithms are usually employed, combined with local search methods.

MAs operation follows closely the operation of the employed evolutionary algorithm. However, after applying the evolutionary operators and evaluating the population, S, a subset, S_{loc}, is selected as initial points for the local search. The obtained solutions replace these individuals in the population, if they improve their function values.

The selection of S_{loc} as well as the frequency of application of the local search is considered a topic of crucial importance in the field of MAs and it is usually problem dependent [26]. The employed local search methods play also an important role in the applicability and efficiency of MAs. While gradient–based local search can be very efficient in well–defined problems with good Mathematical characteristics, stochastic local search is considered more appropriate for problems contaminated by noise and uncertainty, as in most real life applications. Different local search schemes have been used in MAs and results are reported in relative literature [25].

In the proposed memetic algorithm, we utilized the Solis and Wets (SW) [27] local search technique. The main criterion for the selection of this scheme was the preservation of the derivative–free nature of the memetic scheme, as well as its simplicity and flexibility for adaptation on the problem at hand. Solis and Wets proposed several stochastic local search algorithms. In our approach, we used the algorithm reported as "Algorithm 1" in [27].

In the MAs that we investigated on the problems of detecting periodic orbits, PSO and DE were used as the global search component of the memetic algorithm, with SW employed as the local search component. Regarding the frequency of the local search application, two different schemes were considered, namely, a simple scheme that admits local search at each iteration of the algorithm, and a more sophisticated scheme that applies local search only if, after a prespecified number of iterations, the entropy of the population falls under a user–defined value [14]. In both cases, SW was applied on the best positions of the particles (in PSO) or the individuals (in DE) with a prespecified probability [14, 15]. Regarding the implementation difficulty of the algorithms, it clearly depends on the selected local search scheme. PSO and DE are both considered very simple population–based algorithms and they can be implemented in a few lines, as exhibited in [28–

30] for different programming languages. Also, the aforementioned local search algorithms are very simple in implementation, rendering the considered memetic schemes very simple.

4. Experimental results

The following mappings were considered in our experiments:
TEST PROBLEM 1 (TP1) [1, 5, 31] (Hénon 2–dimensional map). This mapping is defined by the following equation:

$$\Phi(X) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 - X_1^2 \end{pmatrix},$$

where α is the rotation angle. We investigated this problem for period $p = 10368$, and $X_1 \in [0.85, 0.87]$, $X_2 \in [0.12, 0.14]$, $\cos\alpha = 0.24$ [1].
TEST PROBLEM 2 (TP2) [32] (Hénon 4–dimensional symplectic map). This mapping describes the effects of a particle's motion through nonlinear magnetic focusing elements of the FODO cell type, and it is defined by the following equation:

$$\begin{pmatrix} \Phi_1(X) \\ \Phi_2(X) \\ \Phi_3(X) \\ \Phi_4(X) \end{pmatrix} = \begin{pmatrix} R(\alpha_1) & \mathcal{O} \\ \mathcal{O} & R(\alpha_2) \end{pmatrix}$$

$$\times \begin{pmatrix} X_1 \\ X_2 + X_1^2 - X_3^2 \\ X_3 \\ X_4 - 2X_1X_3 \end{pmatrix},$$

where $R(\alpha)$, $\mathcal{O}$, are defined as [1]:

$$R(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}, \qquad \mathcal{O} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

and $\alpha_1 = 0.61903$, $\alpha_2 = 0.4152$ are the frequencies [32]. We investigated this problem for period $p = 14606$, and for $X_1 \in [0.26, 0.30]$, $X_2 \in [0.07, 0.11]$, $X_3 \in [0.51, 0.55]$, and $X_4 \in [0.13, 0.17]$.

TEST PROBLEM 3 (TP3) [33] (Driven Duffing Oscillator). This is a continuous conservative dynamical system, defined as

$$\ddot{X} = X - X^3 + \alpha\cos\omega t,$$

where $\alpha = 0.05$ and $\omega = 2$. We investigated this problem for period $p = 663$ and $X \in [0.42, 0.56]$, $\dot{X} \in [-0.05, 0.15]$ [33]. For each test problem, the PSO and DE algorithms described in Section 3 were applied on the mapping within the corresponding search ranges. Our target was to illustrate the performance of these algorithms on different test problems and for objectives functions defined by different norms, as stated in Section 2. For this purpose, we investigated the most common norms, namely, ℓ_2, ℓ_1 and ℓ_∞. More specifically, each algorithm was applied 100 times on each test problem for each norm. At every experiment, the success of the algorithm in achieving a target minimum objective function equal to 10^{-8} as well as the required number of function evaluations, were

Table 2. Results of DE for TP1

Norm	SS	Max	Min	Mean	StD	Suc
ℓ_2	100	42301	10801	24814.4	6552.2	97
	150	90901	19201	47665.0	13473.9	100
ℓ_1	100	53401	10801	24968.7	6489.9	99
	150	81301	21151	46519.2	10121.5	99
ℓ_∞	100	36301	10201	24584.8	6217.1	99
	150	73951	23101	43984.0	11624.8	100
$M\ell_\infty$	100	9464	759	3111.7	1713.8	100

Table 3. Results of PSO for TP2

Norm	SS	Max	Min	Mean	StD	Suc
ℓ_2	100	93000	13900	31480.6	15296.0	98
	150	79200	24900	38134.5	11229.2	100
ℓ_1	100	83800	16400	28944.7	12013.0	94
	150	94350	21900	39392.4	12955.6	99
ℓ_∞	100	76700	16700	31254.7	12951.1	95
	150	88950	18750	37898.4	12451.2	96
$M^E\ell_2$	80	92946	11680	25911.9	15489.4	100

Table 4. Results of DE for TP2

Norm	SS	Max	Min	Mean	StD	Suc
ℓ_2	100	44401	17201	26744.5	5336.9	92
	150	68401	24601	45452.5	8442.5	99
ℓ_1	100	36901	18001	26528.4	4148.6	95
	150	66451	25501	44855.6	8047.4	97
ℓ_∞	100	50801	17601	27904.2	5673.1	93
	150	84001	30001	45696.7	9770.2	94
$M\ell_2$	80	67584	4505	25829.4	13402.4	100

recorded. If the target was not achieved within a budget of 10^5 function evaluations, then the algorithm was terminated recording a failure.

Since the efficacy of a population–based algorithm stems from the interactions among population (swarm) members, it is an open question what the swarm size shall be for a specific problem. In most cases, this cannot be determined but experimentally. In order to investigate the behavior of the proposed approaches under population size scaling, we repeated all experiments for two different population sizes, denoted as SS, that were dependent on the algorithm and the problem at hand, based on our previous experience [12–16].

Moreover, in TP1 and TP2, where PSO and DE needed a significant computational effort to detect the periodic orbits, their memetic counterparts were also applied for each norm. The SW local search algorithm was used in the memetic approaches. As mentioned in Section 3 3.3, two different memetic schemes were considered, namely, an entropy–based scheme, and a simpler one that applies local search at each iteration. For each test problem and algorithm, only the best performing memetic scheme was recorded.

Regarding TP1, PSO was applied for swarm sizes 40 and 60 and the results are reported in Table 1. More specifically, for each norm and swarm size, the maximum, minimum, mean and standard deviation of the required function evaluations, along with the number of successes (out of 100 experiments), are reported. Also, the best performing memetic scheme, which, in this case, is the one with the ℓ_∞ norm and application of the local search at each iteration on each best position with probability 0.1, is reported (and denoted as $M\ell_\infty$). SW was let to perform 100 function evaluations per application. As we observe, although the standard PSO exhibit high success rate, the memetic approach reduces significantly the required number of function evaluations, while retaining its efficiency. As a general trend, higher population size can be associated with better performance.

The corresponding results for the DE algorithm on TP1, are reported in Table 2. However, DE required larger population sizes to achieve success rates similar to PSO. This can be attributed to the greedy nature of the DE implied by its inherent selection that filters the population at each iteration. This characteristic makes DE prone to get stuck in local minima, especially when high accuracy in the solutions is required. Regarding the memetic approach, the same scheme as for PSO exhibited the best performance, reducing significantly the mean number of function evaluations. Overall, higher population sizes in DE correspond to higher number of function evaluations (in contrast to PSO). In TP2, PSO and DE assumed the same population sizes, equal to 100 and 150, with similar performance, as reported in Tables 3 and 4. However, the memetic PSO approach with entropy outperformed the rest memetic PSO schemes (denoted as $M^E\ell_2$ in Table 3). On the other hand, the simple memetic DE approach with application of SW at each iteration and on each individual with probability 0.2, was the best among the different memetic DE schemes. Also, in this case, SW was allowed to perform a maximum of 200 function evaluations, both for PSO and DE. This modification was necessary due to the higher dimensionality of TP2, compared to TP1 where 100 evaluations were sufficient.

Table 5. Results of PSO for TP3

Norm	SS	Max	Min	Mean	StD	Suc
ℓ_2	40	9880	3600	6008.9	1357.1	100
ℓ_1	40	12120	2680	6240.8	1570.8	100
ℓ_∞	40	14160	3320	6541.1	2010.7	97

Table 6. Results of DE for TP3.

Norm	SS	Max	Min	Mean	StD	Suc
ℓ_2	40	8921	1641	3479.8	1303.1	100
ℓ_1	40	8721	1681	3667.8	1419.1	100
ℓ_∞	40	5721	1761	2971.0	860.8	100

Finally, TP3 was solved relatively easier than TP1 and TP2. For this purpose, only a single swarm size equal to 40 was considered, without the need for applying memetic approaches, both for PSO and DE. Regarding the norms, ℓ_2 was the best choice for PSO, while ℓ_∞ was the best for DE. We must notice that the basin of convergence of the deterministic Newton method for the specific orbit is significantly degraded as illustrated in [33].

5. Conclusions

This paper provides an overview of recently proposed approaches for detecting periodic orbits of nonlinear mappings using population–based stochastic optimization algorithms. The transformation of the problem to an equivalent optimization task provides the advantage of exploiting the existence of chaos to the benefit of the employed optimization algorithm, while it makes a wide variety of algorithms available to the user.

Putting comparisons aside, two different algorithms, namely PSO and DE, were illustrated on three typical mappings for the detection of long periodic orbits. Also, in cases where computational burden was relatively heavy, memetic approaches of the two algorithms were applied successfully, extending the established approaches. The results imply that stochastic optimization can be a very useful alternative tool, and it has the advantage of wide applicability, even in cases where traditional gradient–based algorithms fail or cannot be applied straightforwardly, e.g., cases of discontinuous or nondifferentiable mappings, even for long periodic orbits. The required number of function evaluations is in direct relevance with the desired accuracy, while the scaling in higher dimensions does not pose significant computational burden in properly selected memetic schemes.

Also, experiments reveal the heavy dependence of each algorithm's performance on the problem at hand as well as other factors, such as the selected norm and population size. Further research can provide intuition regarding the most appropriate choice per case as well as further aspects of each algorithm's behavior that can help towards the design of more specialized operators for this kind of problems.

References

[1] M. N. Vrahatis, J. Comp. Phys. **119**, 105 (1995).

[2] G. D. Birkhoff, Trans. Amer. Math. Soc. **18**, 199 (1917).

[3] T. C. Bountis and R. H. Helleman, J. Math. Physics **22**, 1867 (1981).

[4] R. H. G. Helleman, in *Statistical Mechanics and Statistical Methods in Theory and Applications*, edited by U. Landman (Plenum, 1977), p. 343.

[5] M. Hénon, Quart. Appl. Math. **27**, 291 (1969).

[6] C. Polymilis, C. Skokos, G. Kollias, G. Servizi, and G. Turchetti, J. Phys. A **33**, 1055 (2000).

[7] C. Skokos, G. Contopoulos, and C. Polymilis, Cel. Mech. Dyn. Astron. **65**, 223 (1997).

[8] M. N. Vrahatis and T. C. Bountis, in *Proc. Int. Conf. Hamiltonian Mechanics, Integrability and Chaotic Behavior*, edited by J. Seimenis (1994), pp. 261–274.

[9] M. N. Vrahatis, T. C. Bountis, and M. Kollmann, Inter. J. Bifurc. Chaos **6**, 1425 (1996).

[10] F. Verhulst, *Nonlinear Differential Equations and Dynamical Systems* (Springer–Verlag, Berlin, 1990).

[11] P. Cvitanović, Physica D **51**, 138 (1991).

[12] C. Skokos, K. E. Parsopoulos, P. A. Patsis, and M. N. Vrahatis, Monthly Notices of the Royal Astronomical Society **359**, 251 (2005).

[13] K. E. Parsopoulos and M. N. Vrahatis, in *Proceedings of the IEEE 2003 Swarm Intelligence Symposium* (Indianapolis (IN), USA, 2003), pp. 34–41.

[14] Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis, in *IEEE 2007 Congress on Evolutionary Computation (CEC'07)* (Singapore, 2007), pp. 2040–2047.

[15] Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis, Annals of Operations Research **156**, 99 (2007).

[16] Y. G. Petalas, K. E. Parsopoulos, E. I. Papageorgiou, P. P. Groumpos, and M. N. Vrahatis, in *IEEE 2007 Swarm Intelligence Symposium (SIS'07)* (Hawaii (HI), USA, 2007), pp. 16–22.

[17] K. E. Parsopoulos and M. N. Vrahatis, IEEE Transactions on Evolutionary Computation **8**, 211 (2004).

[18] C. Skokos, K. E. Parsopoulos, P. A. Patsis, and M. N. Vrahatis, in *6th Hellenic Astronomical Conference* (Athens, Greece, 2003).

[19] J. Kennedy and R. C. Eberhart, in *Proc. IEEE Int. Conf. Neural Networks* (IEEE Service Center, Piscataway, NJ, 1995), vol. IV, pp. 1942–1948.

[20] J. Kennedy and R. C. Eberhart, *Swarm Intelligence* (Morgan Kaufmann Publishers, 2001).

[21] M. Clerc and J. Kennedy, IEEE Trans. Evol. Comput. **6**, 58 (2002).

[22] K. E. Parsopoulos and M. N. Vrahatis, Mathematical and Computer Modelling **46**, 198 (2007).

[23] A. Törn and A. Žilinskas, *Global Optimization* (Springer–Verlag, Berlin, 1989).

[24] R. Storn and K. Price, J. Global Optimization **11**, 341 (1997).

[25] N. Krasnogor and J. Smith, IEEE Trans. Evol. Comput. **9**, 474 (2005).

[26] W. E. Hart, Ph.D. thesis, University of California, San Diego,USA (1994).

[27] F. Solis and R. Wets, Mathematics of Operations Research **6**, 19 (1981).

[28] M. Clerc, *Particle Swarm Optimization* (ISTE Ltd, 2006).

[29] K. E. Parsopoulos and M. N. Vrahatis, Natural Computing **1**, 235 (2002).

[30] `http://www.icsi.berkeley.edu/~storn/code.html`.

[31] C. Polymilis, G. Servizi, and C. Skokos, Cel. Mech. Dyn. Astron. **66**, 365 (1997).

[32] M. N. Vrahatis, H. Isliker, and T. C. Bountis, Inter. J. Bifurc. Chaos **7**, 2707 (1997).

[33] L. Drossos, O. Ragos, M. N. Vrahatis, and T. C. Bountis, Physical Review E **53**, 1206 (1996).