Entropy–Based Memetic Particle Swarm Optimization for Computing Periodic Orbits of Nonlinear Mappings

Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis

***Abstract*— The computation of periodic orbits of nonlinear mappings is very important for studying and better understanding the dynamics of complex systems. Evolutionary algorithms have shown to be an efficient alternative for the computation of periodic orbits in cases where the inherent properties of the problem at hand render gradient–based methods invalid. Such cases usually involve nondifferentiable mappings or poorly behaved partial derivatives. We propose a Memetic Particle Swarm Optimization algorithm that exploits Shannon's information entropy for decision making in swarm level, as well as a probabilistic decision making scheme in particle level, for determining when and where local search is applied. These decisions have a significant impact on the required number of function evaluations, especially in cases where high accuracy is desirable. Experimental results are performed on well–known problems and useful conclusions are derived.**

I. INTRODUCTION

Natural phenomena and complex systems are usually modeled by dynamical systems that involve nonlinear mappings [1]–[14]. The points that are invariant under the mapping are called *fixed points* or *periodic orbits* of the mapping and they are considered very important, since they can provide crucial information regarding its behavior. If

$$\Phi(x) = \big(\Phi_1(x), \ldots, \Phi_n(x)\big)^\top : \mathbb{R}^n \to \mathbb{R}^n,$$

is a nonlinear mapping, then a point $x = (x_1, x_2, \ldots, x_n)^\top \in \mathbb{R}^n$, is a periodic orbit of period p of Φ, if p applications of Φ on x result in the same point, x, i.e.,

$$x = \Phi^p(x) \equiv \underbrace{\Phi(\Phi(\ldots\Phi(x)\ldots))}_{p \text{ times}}.$$

The detection of periodic orbits requires solving the system $x = \Phi^p(x)$, which can be equivalently defined as an optimization problem by considering the objective function

$$f(x) = D\left(\Phi^p(x), x\right), \qquad (1)$$

where $D(\alpha, \beta)$ is a distance measure for the vectors α and β. Typically, common norms such as the ℓ_1, ℓ_2 and ℓ_∞–norms, defined as

$$\|\varepsilon\|_1 = \sum_{i=1}^{n} |\varepsilon_i|,$$

$$\|\varepsilon\|_2 = \left(\sum_{i=1}^{n} |\varepsilon_i|^2\right)^{1/2},$$

$$\|\varepsilon\|_\infty = \max_{1 \leqslant i \leqslant n} |\varepsilon_i|,$$

respectively, are used as the distance measures, with $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^\top$, and $\varepsilon_i = \Phi_i^p(x) - x_i$, for $i = 1, 2, \ldots, n$ [1], [15], [16].

The computation of periodic orbits is considered a challenging problem. Analytical derivation of such points is feasible only for mappings of special types (e.g., polynomials of low degree) and low periods [6]. On the other hand, approximations of such points with gradient–based methods, such as the Newton–family algorithms, are heavily dependent on the initial conditions, as well as on the behavior of the partial derivatives, if they exist, around the fixed point. Evolutionary algorithms have shown to be an efficient alternative in cases where traditional methods fail. Particle Swarm Optimization (PSO) has been applied successfully for tracing periodic orbits of 3D galactic potentials [15] and well–known nonlinear mappings [16], [17].

Memetic Algorithms (MAs) are hybrid optimization algorithms that consist of a global and a local search component [18]–[22]. The first is responsible for the detection of the global minimizers, while the latter is used for more refined local search in the neighborhood of detected potential minimizers. MAs are suitable for numerical optimization problems where high accuracy is required, since the local search component can refine significantly the rough solution set detected by the (usually evolutionary) global component. Recently, Memetic Particle Swarm Optimization (MPSO) variants that combine PSO with different stochastic and deterministic local search schemes, such as Hooke and Jeeves, Solis and Wets, and Random Walk with Directional Exploitation, have been proposed and applied on different applications, outperforming the standard PSO variants [23]–[25].

In this work, we propose a new MPSO algorithm for computing periodic orbits of nonlinear mappings with high accuracy. The new scheme exploits the concept of Shannon's information entropy (SIE) [26], in order to identify search stagnation of the swarm during optimization. Stagnation evokes the local search component, which is applied

The authors are with the Computational Intelligence Laboratory (CI Lab), Department of Mathematics, University of Patras, GR–26110 Patras, Greece and University of Patras Artificial Intelligence Research Center (UPAIRC) (email: {petalas,kostasp,vrahatis}@math.upatras.gr)

1-4244-1340-0/07/$25.00 ©2007 IEEE

probabilistically on the best position of each particle. The performance of the proposed approach is investigated on four widely used mappings for several periods, and it is compared to the standard PSO algorithm.

The rest of the paper is organized as follows: Section II contains the necessary background material, while Section III is devoted to the description of the proposed scheme. Experimental results are reported in Section IV, and the paper closes with conclusions in Section V.

II. Background Material

For completeness purposes, the following two subsections are devoted to the description of MPSO and SIE.

A. Memetic Particle Swarm Optimization

Swarm intelligence is the class of algorithms that model information exchange and communication features of natural swarms (ant colonies, fish shoals etc.), in order to produce intelligent (emergent) behavior of populations consisting of simple agents. PSO belongs to this class of algorithms, which are characterized by proximity, quality, diverse response, stability, and adaptability [27]. Search is performed by a *swarm* of points, called *particles*. The particles are initialized randomly in the search space and they move by stochastic forces that attract them towards the best solutions detected by them as well as their neighbors.

Let $S \subset \mathbb{R}^n$ be the search space, and $\mathbb{S} = \{x_1, x_2, \ldots, x_N\}$ be the swarm that consists of N particles. Each particle is a point in S,

$$x_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})^\top \in S, \qquad i = 1, 2, \ldots, N.$$

The particles are usually initialized randomly and uniformly in S. Besides its position, for each particle, a randomly initialized velocity,

$$v_i = (v_{i_1}, v_{i_2}, \ldots, v_{i_n})^\top, \qquad i = 1, 2, \ldots, N,$$

is also assigned. The best position ever visited by the particle in S is retained in a memory,

$$b_i = (b_{i_1}, b_{i_2}, \ldots, b_{i_n})^\top \in S, \qquad i = 1, 2, \ldots, N.$$

For each particle, a social environment with which it will exchange information, is defined. This environment is called the *neighborhood*, $\mathbb{N}_i \subseteq \mathbb{S}$, of the particle.

The scheme for determining $\mathbb{N}_i$ for each particle is also called *topology* of the neighborhood. The most typical topologies are the *ring* (also called *lbest*) and the *fully connected* (also called *gbest*) topology. In the first, the particles are assumed to lie on a ring, and x_i exchanges information only with its immediate neighbors on the ring, with x_1 being the particle that follows immediately after x_N. Thus, in a ring topology of radius ρ, the neighborhood of the particle x_i is defined as

$$\mathbb{N}_i^\rho = \{x_{i-\rho}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{i+\rho}\}.$$

In the fully connected topology, the whole swarm is considered as the neighborhood of each particle, i.e., $\mathbb{N}_i = \mathbb{S}$ for all $i = 1, 2, \ldots, N$.

Let g_i denote the index of the particle that attained the best position among all the particles in the neighborhood of x_i, i.e.,

$$f(p_{g_i}) \leqslant f(p_k), \qquad \text{for all } k \text{ such that } x_k \in \mathbb{N}_i,$$

and t to be the iteration counter. Then, the swarm is updated using the equations [28],

$$\begin{aligned} F_{i_j}(t+1) &= \varphi_1\Big(b_{i_j}(t) - x_{i_j}(t)\Big) + \varphi_2\Big(b_{g_{i_j}}(t) - x_{i_j}(t)\Big), \\ v_{i_j}(t+1) &= \chi\left[v_{i_j}(t) + F_{i_j}(t+1)\right], \\ x_{i_j}(t+1) &= x_{i_j}(t) + v_{i_j}(t+1), \\ i &= 1, 2, \ldots, N, \qquad j = 1, 2, \ldots, n, \end{aligned} \tag{2}$$

where χ is the *constriction factor* that controls the magnitude of the velocity, and φ_1, φ_2, are two positive random values uniformly distributed within ranges $[0, c_1]$ and $[0, c_2]$, respectively. The parameters c_1 and c_2 control the cognitive and social effect on the velocities, respectively.

The stability analysis of Clerc and Kennedy [28] suggested that χ is obtained from the analytical formula,

$$\chi = \frac{2\kappa}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|},$$

where $\varphi = c_1 + c_2$. The values received for $\varphi > 4$ and $\kappa = 1$ are considered the most common settings of χ due to their good average performance [28].

MPSO is based on the operation of the standard PSO described above. However, a local search scheme is incorporated to the algorithm and applied under specific conditions, which are usually problem–dependent. Critical issues in MAs are the point and frequency of application of the local search, as well as the available budget of function evaluations and the type of the local search algorithm. Hart [20] investigated the effect of local search frequency in combination with Genetic Algorithms in continuous optimization problems. He also studied fitness–based and distribution–based techniques for adapting the probability of applying local search at each individual.

In recently proposed MPSO schemes, local search was applied on best positions of the particles in a probabilistic way, based on a user–defined threshold [23]–[25]. The procedure of MPSO is outlined in the following pseudocode:

Initialize swarm, velocities and best positions
While (stopping criterion not satisfied) **Do**
  Update particles, velocities and best positions
  Select best positions for local search
  Apply local search
  If (improvement attained) **Update** best positions
End Do

In all cases, local search was applied always after a fixed number of iterations, since the main goal was a balanced search that combines the exploration ability of PSO with the exploitation properties of the employed local search schemes. However, in cases such as the detection of periodic orbits, high accuracy is usually required.

Applying the general scheme described above could result in an excessive number of function evaluations, uniformly spent over the iterations of the algorithm. Therefore, the proportion of a limited budget of function evaluations that would be left for refinement of the rough solutions detected through PSO, would be very small. For this purpose, a diversity measure that could indicate, in swarm level, the most proper time for applying local search could be very useful. Such a diversity measure is the SIE, which is described in the next subsection.

B. Shannon's Information Entropy

Shannon's information entropy (SIE) [26] has been used as a diversity measure for populations in Genetic Programming [29], [30]. For a population, P, divided in k phenotype classes, SIE is defined as

$$\mathrm{SIE}(P) = -\sum_{k} p_k \log p_k,$$

where p_k is the proportion of P occupied by partition k at a given time [30].

The entropy principle has been combined with a mechanism based on natural immune system and applied for preserving diversity in the population of a multiobjective evolutionary algorithm [31]. Further work on parallel memetic schemes combined with entropy–based techniques for determining the initial conditions of the local search is reported in [32].

SIE represents the amount of chaos in a system. Large values of entropy correspond to small values of p_k, i.e., each partition has a significant number of individuals. On the other hand, small values of entropy correspond to larger values of p_k, i.e., a significant number of individuals is concentrated in few partitions. Therefore, high entropy indicates higher diversity in the population, in analogy with physical systems [26]. The concept of SIE is adopted for fitting to our periodic orbit detection problems, as it is analyzed in the next section.

III. Entropy–Based Memetic Particle Swarm Optimization

The development of the proposed *Entropy–Based Memetic Particle Swarm Optimization* (E–MPSO) was motivated by the difficulty of the standard PSO in detecting periodic orbits with high accuracy. Experiments on different mappings have shown that for accuracy greater than 10^{-10}, PSO is prone to search stagnation or requires an excessive number of function evaluations. However, we were also interested in avoiding disturbing PSO's dynamic in cases where it performed satisfactorily. Thus, a mechanism for making decisions in swarm level regarding the application of local search was needed. A decision making procedure in particle level was also needed in order to select the individuals that would constitute the initial conditions of the local search.

SIE has been used as a measure of diversity in evolutionary algorithms, providing information regarding the spread of the individuals' function values. Thus, monitoring its value during the optimization procedure provides information regarding the population's behavior. High values of SIE correspond to widely spread function values, while smaller values indicate similar function values of the individuals. Also, a rapidly changing value of SIE is an indication of rapidly changing diversity of the population, while slight changes of SIE indicate that the relative differences among function values of the population remain almost unchanged, an effect that also characterizes search stagnation.

Therefore, SIE was selected as the core of the procedure for deciding in swarm level regarding the application or not of local search. More specifically, the changes in the value of SIE are monitored in equidistant intervals (e.g., every k iterations), and, if the difference of the current and previous value is smaller than a user–defined threshold value, then the local search component of the algorithm is evoked.

Let $\mathbb{S}$ be a swarm of size N. Then, the SIE of $\mathbb{S}$ at iteration t, is computed by

$$\mathrm{SIE}_t(\mathbb{S}) = -\sum_{i=1}^{N} P_i(t) \log P_i(t), \qquad (3)$$

where,

$$P_i(t) = f\left(b_i(t)\right) \Big/ \sum_{i=1}^{N} f\left(b_i(t)\right),$$

i.e., $P_i(t)$ is the proportion of the function value of ith particle's best position, b_i, at iteration t. If

$$|\mathrm{SIE}_t(\mathbb{S}) - \mathrm{SIE}_{t-k}(\mathbb{S})| \leqslant T_{\mathrm{SIE}},$$

where T_{SIE} is a user–defined threshold value, then the local search component is evoked. However, not all best positions of the swarm will be used as initial conditions for local search. Since the function values of the best positions will be close as indicated by the value of SIE, a randomized, non–elitist selection of best positions is performed. Thus, for each best position, b_i, with $i = 1, 2, \ldots, N$, a random value, R_{b_i} uniformly distributed within $[0, 1]$, is drawn, and, if

$$R_{b_i} \leqslant P_s,$$

where P_s is a user–defined selection probability, then b_i is used as an initial condition for local search, otherwise it is ignored. The non–elitism can prevent from premature convergence to local minima, while the selection pressure imposed by the user–defined threshold prevents from excessively large numbers of required function evaluations. A pseudocode of the E–MPSO scheme is reported in Table I.

IV. Experimental Results

A. The Considered Test Problems

The mappings that were considered in our experiments are:

Test Problem 1 [1], [6] (Hénon 2–dimensional map) This mapping is 2–dimensional and defined by the following equation:

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

TABLE I

PSEUDOCODE OF THE E–MPSO SCHEME.

```
Input: Swarm S, Size N, Parameters T_SIE, P_s, k
Set t = 0 and SIE_prev = SIE_t(S)
While (stopping criterion not satisfied) Do
    Set t = t + 1
    Update S using Eq. (2)
    If (mod(t, k) = 0) Then
        Compute SIE_t(S)
        If (|SIE_t(S) − SIE_prev| ⩽ T_SIE) Then
            For i = 1 : N Do
                Draw a random number R_{b_i} ∈ [0, 1]
                If (R_{b_i} ⩽ P_s) Then
                    Apply local search on b_i.
                    Update b_i if improvement achieved.
                End If
            End For
        End If
        Set SIE_prev = SIE_t(S)
    End If
End While
```

$$\begin{cases} \Phi_1(x) = x_1 \cos\alpha - (x_2 - x_1^2) \sin\alpha, \\ \Phi_2(x) = x_1 \sin\alpha + (x_2 - x_1^2) \cos\alpha, \end{cases}$$

where $\alpha \in [0, \pi]$ is the rotation angle. The case of $\cos\alpha = 0.24$ was considered for periods $p = 5$ and $p = 17$ within $[-1, 1]^2$.

TEST PROBLEM 2 [33] (Standard Map) This is also a 2–dimensional mapping. It is discontinuous and defined by the following equation:

$$\begin{cases} \Phi_1(x) = \left(x_1 + x_2 - \frac{k}{2\pi}\sin(2\pi x_1)\right) \bmod \frac{1}{2}, \\ \Phi_2(x) = \left(x_2 - \frac{k}{2\pi}\sin(2\pi x_1)\right) \bmod \frac{1}{2}, \end{cases}$$

where $k = 0.9$, and

$$y \bmod \frac{1}{2} = \begin{cases} (y \bmod 1) - 1, & \text{if } (y \bmod 1) > \frac{1}{2}, \\ (y \bmod 1) + 1, & \text{if } (y \bmod 1) < -\frac{1}{2}, \\ (y \bmod 1), & \text{otherwise}. \end{cases}$$

This mapping was considered also for periods $p = 5$ and $p = 17$ within $[-1, 1]^2$.

TEST PROBLEM 3 [1] (Hénon 4–dimensional symplectic map) This 4–dimensional map is an extension of the Hénon 2D map to the complex case:

$$\begin{pmatrix} \Phi_1(x) \\ \Phi_2(x) \\ \Phi_3(x) \\ \Phi_4(x) \end{pmatrix} = \begin{pmatrix} R(\alpha) & \mathcal{O} \\ \mathcal{O} & R(\alpha) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 - x_1^2 + x_3^2 \\ x_3 \\ x_4 - 2x_1x_3 \end{pmatrix},$$

where α is the rotation angle, and $R(\alpha)$, $\mathcal{O}$, are defined as [1]:

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

The case of $\alpha = \cos^{-1}(0.24)$ was considered, for periods $p = 5$ and $p = 11$ within $[-3, 3]^4$.

TABLE II

PARAMETER SETTINGS FOR THE E–MPSO SCHEME.

Parameter	Value(s)
χ	0.729
c_1, c_2	2.05
ρ	1
Swarm size	20 up to 300
Overall function evaluations	15×10^5, 20×10^5
Function evaluations per local search	100
Accuracy	10^{-16}
T_{SIE}	10^{-3}
k	10
P_s	0.1, 0.2, 0.3

TEST PROBLEM 4 [34], [35] This is a 6–dimensional map, and it represents a case of the standard maps studied by Kantz and Grassberger [35]. It is defined by the following equations:

$$\begin{cases} x_1' = x_1 + x_2' \\ x_2' = x_2 + \frac{K}{2\pi}\sin(2\pi x_1) - \frac{\beta}{2\pi}\{\sin[2\pi(x_5 - x_1)] + \\ \quad \sin[2\pi(x_3 - x_1)]\} \\ x_3' = x_3 + x_4' \\ x_4' = x_4 + \frac{K}{2\pi}\sin(2\pi x_3) - \frac{\beta}{2\pi}\{\sin[2\pi(x_1 - x_3)] + \\ \quad \sin[2\pi(x_5 - x_3)]\} \\ x_5' = x_5 + x_6' \\ x_6' = x_6 + \frac{K}{2\pi}\sin(2\pi x_5) - \frac{\beta}{2\pi}\{\sin[2\pi(x_3 - x_5)] + \\ \quad \sin[2\pi(x_1 - x_5)]\} \end{cases} \pmod 1$$

All variables are given (mod 1), so $x_i \in [0, 1)$, for $i = 1, \ldots, 6$. For $\beta = 0$, the map gives three uncoupled standard maps, while for $\beta \neq 0$ the maps are coupled and influence each other. In our experiments, $\beta = K = 1$, and the periods $p = 3$ and $p = 7$ were considered.

B. Algorithmic Settings

Our E–MPSO algorithm consisted of the standard PSO with ring topology (lbest) as the global search component, and the stochastic Solis and Wets (SW) [36] algorithm as the local search component. SW has been used in successful memetic schemes [24], [25]. Whenever local search was applied, a budget of 100 function evaluations was available to the SW algorithm. Naturally, detecting a point that improves the initial condition, seizes the local search procedure and returns the improved point, without exhausting this budget.

Regarding PSO, the neighborhood radius was $\rho = 1$, since it promotes exploration. Also, the default set of parameters, $\chi = 0.729$, $c_1 = c_2 = 2.05$ was used [28]. The corresponding optimization problem for each mapping was produced using Eq. (1) with the squared ℓ_2–norm. Therefore, the global minimum was always equal to 0. The desired accuracy for detecting this global minimum was equal to 10^{-16} for all test problems. The available number of function evaluations was problem–dependent, due to the different computational effort required for different test problems. Thus, 15×10^5 function evaluations were available for Test Problems (TP) 1–3, while 20×10^5 function evaluations were available for the case $p = 7$ of TP 4 (which was the problem

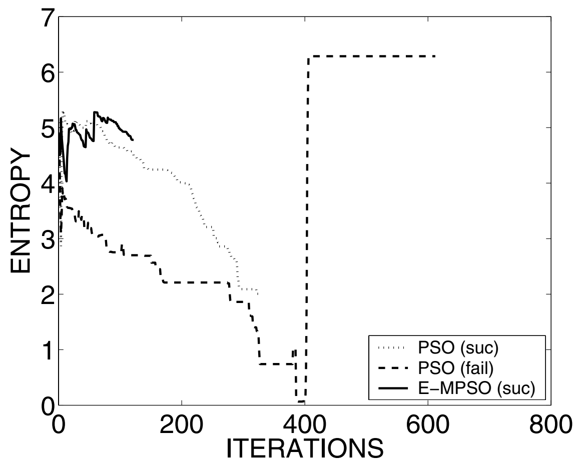

Fig. 1. Evolution of SIE values for successful and failure cases of PSO (dotted and dashed lines, respectively), and E–MPSO (solid line).

TABLE III

RESULTS FOR TP1 AND PERIOD $p = 5$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 20, 30, AND 40.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
20	Suc	100	100	100	98
	Mean	**4453.3**	4537.3	4603.8	5311.8
	StD	1864.3	**1576.5**	1780.6	5095.2
	Min	2580.0	2814.0	2740.0	2620.0
	Max	14343.0	13151.0	13139.0	35160.0
30	Suc	100	100	100	100
	Mean	5891.7	**5686.8**	5983.3	6245.1
	StD	1341.4	**1132.0**	1524.3	2072.6
	Min	3600.0	3690.0	3660.0	3780.0
	Max	14557.0	10919.0	13068.0	16680.0
40	Suc	100	100	100	100
	Mean	7230.9	7564.9	**7209.2**	7487.6
	StD	**1203.7**	2045.9	1454.4	1475.7
	Min	5200.0	4760.0	4960.0	5200.0
	Max	13341.0	19567.0	13035.0	17440.0

with the highest dimension). Regarding the SIE decision parameter, we monitored its value every $k = 10$ iterations after the first 20 iterations of the algorithm where PSO was let to perform global search solely. Two consecutive values of SIE that differ less than $T_{\mathrm{SIE}} = 10^{-3}$, triggered the local search component of E–MPSO.

In order to investigate the performance scaling of the algorithm, different swarm sizes as well as probabilities of selection for local search, P_s, were considered for each test problem and period. Thus, the swarm size varied from 20 up to 300, while P_s assumed the values 0.1, 0.2 and 0.3 that correspond to acceptance probability of 10% up to 30% of a best position for serving as starting point for the SW algorithm. All parameter values are summarized in Table II.

C. Analysis of the Results

For each test problem, 100 experiments for E–MPSO and standard PSO were conducted, for different swarm sizes and selection probabilities, P_s. Figure 1 illustrates the values of SIE for a successful and an unsuccessful experiment for the standard PSO (dotted and dashed lines, respectively) as well as for the (successful) E–MPSO (solid line). It is obvious that in successful experiments SIE assumes increasing values in the first iterations, and declines slowly, while in unsuccessful experiments, it decreases rapidly and finally takes a high, fixed value that remains unchanged. The existence of such plateaus has been correlated with search stagnation over local minima [30].

The number of successes in achieving the solution with the desired accuracy was monitored for each algorithm. For the successful experiments, the mean, standard deviation, minimum and maximum number of required function evaluations were recorded. The corresponding statistics are reported in Tables III–X. The smallest values of mean and standard deviation per case are boldfaced in the tables for the completely successful algorithms (i.e., with 100 successes).

For TP1 and period $p = 5$, E–MPSO outperformed standard PSO in almost all cases. Smaller values of P_s were more efficient for small swarm sizes. For larger swarm sizes, standard PSO was more competitive. More specifically, for swarm size equal to 20, E–MPSO always outperformed PSO, with $P_s = 0.1$ exhibiting the best performance. The improvement in the number of the required number of function evaluations lied between 13% and 16% when E–MPSO was used. However, the case of $P_s = 0.2$ was the most promising regarding its robustness, exhibiting the smallest standard deviation. Increasing swarm size to 30, the improvement gained by using E–MPSO lied between 4% and 9%, with $P_s = 0.2$ being the most efficient and robust case. PSO became more competitive when 40 particles were used, outperforming the $P_s = 0.2$ case of E–MPSO. However, it was still inferior than the other two cases, with $P_s = 0.3$ being the most efficient, but $P_s = 0.1$ being the most robust.

The case of period $p = 17$ for TP1, required higher swarm sizes. Again, E–MPSO was superior to standard PSO, which never attained 100 successful experiments. Due to the large number of particles, higher values of P_s proved to be more efficient and robust. The improvement gained by using E–MPSO was up to 36%, which corresponds to more than 66000 function evaluations.

Similar remarks can be made for TP2. Larger swarm sizes favor higher values of P_s. E–MPSO outperformed PSO in all cases, improving its performance from 9% up to 44% for period $p = 5$, and from 20% up to 56% for $p = 17$. Naturally, increasing the swarm size resulted in improved performance of PSO, which approaches E–MPSO.

The motif does not change for the rest of the test problems, TP3 and TP4. E–MPSO always outperforms PSO. In some cases, such as in TP3 for period $p = 11$ and swarm size equal to 200, the mean number of function evaluations required by PSO is smaller than the best of E–MPSO, but it corresponds

TABLE IV

RESULTS FOR TP1 AND PERIOD $p = 17$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 100, 200, AND 300.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
100	Suc	100	100	100	95
	Mean	133241.9	126937.4	**117011.3**	183665.3
	StD	98102.5	93325.3	**85997.3**	162501.3
	Min	19700.0	20500.0	16100.0	21900.0
	Max	520799.0	434137.0	519051.0	740000.0
200	Suc	100	100	100	97
	Mean	185822.7	152010.6	**150517.2**	195183.5
	StD	123751.1	**92417.2**	97912.6	206799.0
	Min	33800.0	41000.0	38600.0	29600.0
	Max	540157.0	421632.0	412707.0	1292000.0
300	Suc	100	100	100	99
	Mean	175481.6	**165381.1**	203563.5	242993.9
	StD	110062.3	**104915.0**	117223.8	201048.1
	Min	61675.0	50700.0	52800.0	53400.0
	Max	598460.0	531558.0	618655.0	1178700.0

TABLE V

RESULTS FOR TP2 AND PERIOD $p = 5$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 20, 30, AND 40.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
20	Suc	100	100	100	100
	Mean	9144.5	**8111.3**	8179.2	14542.8
	StD	8075.8	5243.4	**4828.9**	34704.1
	Min	3560.0	3440.0	3440.0	2780.0
	Max	64037.0	45169.0	39687.0	306020.0
30	Suc	100	100	100	100
	Mean	**8339.1**	9741.4	8823.0	11068.8
	StD	**2845.2**	4232.9	2881.3	14320.2
	Min	5040.0	5370.0	4680.0	3930.0
	Max	25601.0	25853.0	17738.0	113430.0
40	Suc	100	100	100	100
	Mean	**10842.4**	11222.4	11574.7	11871.2
	StD	4130.1	**3278.1**	7196.3	8751.5
	Min	6120.0	6480.0	6320.0	5560.0
	Max	30054.0	24369.0	70543.0	75800.0

TABLE VI

RESULTS FOR TP2 AND PERIOD $p = 17$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 100, 200, AND 300.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
100	Suc	100	100	100	91
	Mean	61863.9	**55071.3**	62832.8	126150.5
	StD	69939.8	**33270.4**	53974.2	198864.0
	Min	14500.0	17000.0	16400.0	15000.0
	Max	642553.0	174498.0	466300.0	935100.0
200	Suc	100	100	100	98
	Mean	**82830.5**	89506.1	83483.6	103026.5
	StD	58729.9	55183.9	**45847.0**	113954.9
	Min	28400.0	26800.0	34600.0	26000.0
	Max	272460.0	275156.0	231589.0	850000.0
300	Suc	100	100	100	100
	Mean	100294.6	103322.5	**86208.8**	108189.0
	StD	62071.8	55134.3	**36464.8**	89519.3
	Min	29824.0	45000.0	35700.0	33600.0
	Max	329459.0	311971.0	265631.0	703800.0

TABLE VII

RESULTS FOR TP3 AND PERIOD $p = 5$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 40, 60, AND 80.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
40	Suc	100	100	100	97
	Mean	**15677.7**	17209.2	16263.2	15513.4
	StD	**4339.9**	6013.5	5741.4	4752.4
	Min	10080.0	10520.0	10560.0	9840.0
	Max	30200.0	43823.0	51418.0	38200.0
60	Suc	100	100	100	99
	Mean	22159.2	21583.5	**21532.1**	22616.4
	StD	7322.4	4864.6	**4152.0**	15965.4
	Min	14460.0	16620.0	14940.0	14220.0
	Max	66118.0	48131.0	39129.0	171060.0
80	Suc	100	100	100	100
	Mean	26936.1	**26770.3**	26953.8	27306.4
	StD	**4618.3**	5337.7	4627.9	4879.1
	Min	20800.0	20320.0	20320.0	19920.0
	Max	54847.0	58761.0	46462.0	54000.0

to lower success rate (equal to 98%). Thus, E–MPSO may need a slightly higher number of function evaluations, but it is counterbalanced by increased efficiency. The same case of TP3 constitutes an exception also for the general trend observed in the other cases, i.e., the increased efficiency of schemes with higher values of P_s for higher swarm sizes. Instead, we observe that smaller values of P_s tend to perform better.

In general, besides some very rough trends, there are no patterns that can be observed in the reported results, regarding the most efficient E–MPSO scheme, in spite of its clear superiority against standard PSO. This can be attributed to the dynamic nature of the proposed algorithm. SIE is monitored during the optimization at each experiment independently, and the local search component is evoked based on the evidence up to that moment. Thus, the dynamic of the memetic algorithm depends solely on the specific state of the swarm, i.e., the progress of the optimization procedure so far.

V. CONCLUSIONS

We proposed a new, Entropy–based Memetic Particle Swarm Optimization (E–MPSO) for tackling the problem of detecting periodic orbits of nonlinear mappings with high accuracy. The algorithm employs Shannon's information entropy (SIE) for deciding when the Solis and Wets local search component of the algorithm shall be applied (decision at swarm level), as well as a probabilistic scheme for random selection of the corresponding initial conditions for the local search.

TABLE VIII

RESULTS FOR TP3 AND PERIOD $p = 11$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 100, 200, AND 300.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
100	Suc	100	100	100	87
	Mean	**93865.2**	98133.4	102712.4	156218.4
	StD	**47690.4**	52120.4	53818.5	211521.0
	Min	35844.0	28208.0	41286.0	29900.0
	Max	352981.0	339039.0	295876.0	1376400.0
200	Suc	100	100	100	98
	Mean	**147676.5**	153337.6	155330.6	145410.2
	StD	92714.4	80430.2	**63007.2**	69461.2
	Min	70200.0	68706.0	66800.0	61000.0
	Max	857803.0	639616.0	354961.0	399400.0
300	Suc	100	100	100	100
	Mean	194549.5	**183985.0**	196788.7	186918.0
	StD	74741.9	70525.6	**64566.9**	67213.6
	Min	87300.0	99300.0	99600.0	88800.0
	Max	553798.0	509133.0	398680.0	451800.0

TABLE IX

RESULTS FOR TP4 AND PERIOD $p = 3$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 100, 200, AND 300.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
100	Suc	100	100	100	96
	Mean	175189.6	177029.0	**158030.2**	299611.5
	StD	102221.6	116318.7	**89048.4**	293947.2
	Min	44386.0	52400.0	37381.0	48700.0
	Max	568018.0	737581.0	418513.0	1444000.0
200	Suc	100	100	100	100
	Mean	**224235.6**	256251.8	234386.2	277860.0
	StD	**95625.8**	141331.7	107981.4	193392.2
	Min	87000.0	91000.0	80884.0	90200.0
	Max	549585.0	982841.0	572151.0	1294600.0
300	Suc	100	100	100	100
	Mean	316095.1	317639.9	**305069.6**	359019.0
	StD	141522.2	140795.3	**133073.6**	183451.2
	Min	117609.0	135611.0	104700.0	95100.0
	Max	911226.0	899623.0	815164.0	1034100.0

The algorithm was applied on widely used test problems, with very promising results. E–MPSO outperformed PSO and has been shown to be a good alternative when high accuracy is desirable. Also, the dynamic nature of the algorithm is reflected to the reported results. Future work will include further application of E–MPSO on problems where high accuracy is crucial, as well as alternative local search and selection schemes.

ACKNOWLEDGMENT

The Project is co–funded by the European Social Fund and National Resources – (EPEAEK-II) ARCHIMIDIS.

TABLE X

RESULTS FOR TP4 AND PERIOD $p = 7$. THE NUMBER OF SUCCESSES, MEAN, STANDARD DEVIATION, MINIMUM AND MAXIMUM NUMBER OF REQUIRED FUNCTION EVALUATIONS ARE REPORTED FOR E–MPSO AND DIFFERENT VALUES OF P_s, AS WELL AS FOR THE STANDARD PSO, FOR SWARM SIZES EQUAL TO 100, 200, AND 300.

Swarm Size		E–MPSO (P_s) 0.1	0.2	0.3	Standard PSO
100	Suc	100	100	100	87
	Mean	385947.8	334105.2	**332503.9**	472089.7
	StD	318103.4	247310.6	**244096.3**	359761.4
	Min	83119.0	81763.0	83441.0	69200.0
	Max	1834338.0	1201692.0	1954913.0	1455300.0
200	Suc	100	100	100	93
	Mean	489598.9	431093.0	**414523.5**	501587.1
	StD	243055.4	**211083.7**	221468.9	302554.3
	Min	117902.0	134232.0	134198.0	131000.0
	Max	1325777.0	1219383.0	1421575.0	1971800.0
300	Suc	100	100	100	96
	Mean	560497.1	566315.7	**522622.9**	661287.5
	StD	276182.8	264533.5	**235228.7**	321303.0
	Min	174635.0	193818.0	187773.0	225900.0
	Max	1849637.0	1367869.0	1371673.0	1674900.0

REFERENCES

[1] M. N. Vrahatis, "An efficient method for locating and computing periodic orbits of nonlinear mappings," *J. Comp. Phys.*, vol. 119, pp. 105–119, 1995.

[2] G. D. Birkhoff, "Dynamical systems with two degrees of freedom," *Trans. Amer. Math. Soc.*, vol. 18, pp. 199–300, 1917.

[3] T. C. Bountis and R. H. Helleman, "On the stability of periodic orbits of two–dimensional mappings," *J. Math. Physics*, vol. 22, no. 9, p. 1867, 1981.

[4] J. M. Greene, "A method for determining a stochastic transition," *J. Math. Physics*, vol. 20, pp. 1183–1201, 1979.

[5] R. H. G. Helleman, "On the iterative solution of a stochastic mapping," in *Statistical Mechanics and Statistical Methods in Theory and Applications*, U. Landman, Ed. Plenum, 1977, p. 343.

[6] M. Hénon, "Numerical study of quadratic area–preserving mappings," *Quart. Appl. Math.*, vol. 27, pp. 291–311, 1969.

[7] C. Polymilis, C. Skokos, G. Kollias, G. Servizi, and G. Turchetti, "Bifurcations of beam–beam like maps," *J. Phys. A*, vol. 33, pp. 1055–1064, 2000.

[8] C. Skokos, G. Contopoulos, and C. Polymilis, "Structures in the phase space of a four dimensional symplectic map," *Cel. Mech. Dyn. Astron.*, vol. 65, pp. 223–251, 1997.

[9] M. N. Vrahatis and T. C. Bountis, "An efficient method for computing periodic orbits of conservative dynamical systems," in *Proceedings of the International Conference on Hamiltonian Mechanics, Integrability and Chaotic Behavior*, J. Seimenis, Ed., 1994, pp. 261–274.

[10] M. N. Vrahatis, T. C. Bountis, and M. Kollmann, "Periodic orbits and invariant surfaces of 4–D nonlinear mappings," *Inter. J. Bifurc. Chaos*, vol. 6, pp. 1425–1437, 1996.

[11] F. Verhulst, *Nonlinear Differential Equations and Dynamical Systems*. Berlin: Springer–Verlag, 1990.

[12] M. N. Vrahatis, G. Servizi, G. Turchetti, and T. C. Bountis, "A procedure to compute the fixed points and visualize the orbits of a 2D map," European Organization of Nuclear Research, CERN, Tech. Rep. SL/93–06 (AP), 1993.

[13] 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.

[14] G. Servizi, D. Bortolotti, E. Todesco, M. Giovannozzi, and M. N. Vrahatis, "GIOTTO: a code for the nonlinear–analysis of area–preserving mappings," *International Journal of Modern Physics C–Physics and Computers*, vol. 6, no. 5, pp. 651–661, 1995.

[15] C. Skokos, K. E. Parsopoulos, P. A. Patsis, and M. N. Vrahatis, "Particle swarm optimization: An efficient method for tracing periodic orbits in 3D galactic potentials," *Monthly Notices of the Royal Astronomical Society*, vol. 359, pp. 251–260, 2005.

[16] K. E. Parsopoulos and M. N. Vrahatis, "Computing periodic orbits of nondifferentiable/discontinuous mappings through particle swarm optimization," in *Proceedings of the IEEE 2003 Swarm Intelligence Symposium*, Indianapolis (IN), USA, 2003, pp. 34–41.

[17] ——, "Computing periodic orbits of nonlinear mappings through particle swarm optimization," in *Proceedings of the 4th GRACM Congress on Computational Mechanics*, Patras, Greece, 2002.

[18] R. Dawkins, *The Selfish Gene*. Oxford University Press, New York, 1976.

[19] P. Moscato, "Memetic algorithms: A short introduction," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London: McGraw–Hill, 1999, pp. 219–235.

[20] W. E. Hart, "Adaptive global optimization with local search," Ph.D. dissertation, University of California, San Diego,USA, 1994.

[21] N. Krasnogor, "Studies on the theory and design space of memetic algorithms," Ph.D. dissertation, University of the West of England, Bristol, United Kingdom, 2002.

[22] M. W. S. Land, "Evolutionary algorithms with local search for combinatorical optimization," Ph.D. dissertation, University of California, San Diego,USA, 1998.

[23] Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis, "Memetic particle swarm optimization," *Annals of Operations Research*, accepted for publication.

[24] Y. G. Petalas, E. I. Papageorgiou, K. E. Parsopoulos, P. P. Groumpos, and M. N. Vrahatis, "Fuzzy cognitive maps learning using memetic algorithms," in *Proceedings of the International Conference of "Computational Methods in Sciences and Engineering" (ICCMSE 2005)*, 2005, pp. 1420–1423.

[25] ——, "Enhanced learning in fuzzy simulation models using memetic particle swarm optimization," in *IEEE 2007 Swarm Intelligence Symposium (SIS'07)*, Hawaii (HI), USA, 2007, pp. 16–22.

[26] C. E. Shannon, *The Mathematical Theory of Communication*. University of Illinois Press, 1964.

[27] M. M. Millonas, "Swarms, phase transitions, and collective intelligence," in *Computational Intelligence: A Dynamic System Perspective*, M. Palaniswami, Y. Attikiouzel, R. Marks, D. Fogel, and T. Fukuda, Eds. Piscataway, NJ: IEEE Press, 1994, pp. 137–151.

[28] M. Clerc and J. Kennedy, "The particle swarm–explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, 2002.

[29] E. K. Burke, S. Gustafson, and G. Kendall, "Diversity in genetic programming: An analysis of measures and correlation with fitness," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 1098–1107, 2004.

[30] J. P. Rosca, "Entropy–driven adaptive representation," in *Proceedings of the Workshop on Genetic Programming: From Theory to Real–World Applications*, J. P. Rosca, Ed., Tahoe City, California, USA, 1995, pp. 23–32.

[31] X. Cui, M. Li, and T. Fang, "Study of population diversity of multiobjective evolutionary algorithm based on immune and entropy principles," in *Proc. IEEE 2001 Congress on Evolutionary Computation*, vol. 2, 2001, pp. 1316–1321.

[32] J. Tang, M. H. Lim, and Y. S. Ong, "Adaptation for parallel memetic algorithm based on population entropy," in *Proceedings of GECCO 2006*. Seattle, WA, USA: ACM, 2006, pp. 575–582.

[33] S. N. Rasband, *Chaotic Dynamics of Nonlinear Systems*. New York: Wiley, 1990.

[34] C. Skokos, "Alignment indices: A new, simple method for determining the ordered or chaotic nature of orbits," *J. Phys. A*, vol. 34, pp. 10 029–10 043, 2001.

[35] H. Kantz and P. Grassberger, "Internal Arnold diffusion and chaos thresholds in coupled symplectic maps," *J. Phys. A*, vol. 21, pp. 127–133, 1988.

[36] F. Solis and R. Wets, "Minimization by random search techniques," *Mathematics of Operations Research*, vol. 6, pp. 19–30, 1981.