# Particle Swarm Optimization with Dynamic Topology and Conservation of Evaluations

**Carlos M. Fernandes, Juan L.J. Laredo, Juan Julián Merelo, Carlos Cotta and Agostinho C. Rosa**

**Abstract** This paper investigates a Particle Swarm with dynamic topology and a conservation of evaluations strategy. The population is structured on a 2-dimensional grid of nodes, through which the particles interact and move according to simple rules. As a result of this structure, each particle's neighbourhood degree is time-varying. If at given time step *a* particle *p* has no neighbours except itself, *p* is not evaluated until it establishes at least one link to another particle. A set of experiments demonstrates that the dynamics imposed by the structure provides a consistent and stable behaviour throughout the test set when compared to standard topologies, while the conservation of evaluations significantly reduces the convergence speed of the algorithm. The working mechanisms of the proposed structure are very simple and, except for the size of the grid, they do not require parameters and tuning.

**Keywords** Particle swarm optimization • Population structure • Dynamic topologies • Swarm intelligence

C.M. Fernandes (✉) · A.C. Rosa
Institute for Systems and Robotics (ISR/IST), LARSyS, Instituto Superior Técnico,
University of Lisbon, Lisbon, Portugal
e-mail: cfernandes@laseeb.org

A.C. Rosa
e-mail: acrosa@laseeb.orgjuanlu

C.M. Fernandes · J.J. Merelo
Departamento de Arquitectura y Tecnología de Computadores,
University of Granada, Granada, Spain
e-mail: jmerelo@geneura.ugr.es

J.L.J. Laredo
LITIS, University of Le Havre, Le Havre, France
e-mail: juanlu.jimenez@gmail.com

C. Cotta
Lenguages y Ciencias de la Computacion, University of Malaga, Malaga, Spain
e-mail: ccottap@lcc.uma.es

© Springer International Publishing Switzerland 2016

J.J. Merelo et al. (eds.), *Computational Intelligence*,
Studies in Computational Intelligence 620, DOI 10.1007/978-3-319-26393-9_7

# 1 Introduction

The Particle Swarm Optimization (PSO) algorithm is a population-based meta-heuristic for binary and real-valued function optimization inspired by the swarm and social behavior of organisms in bird flocks and fish schools [4]. The optimization is performed by a swarm of candidate solutions, called particles, which move around the problem's search space guided by mathematical rules that define their velocity and position at each time step. Each particle's velocity vector is influenced by its best known position and by the best known positions of its neighbors. The neighborhood of each particle—and consequently the flow of information through the population—is defined a priori by the population topology.

The reason why the swarm is interconnected is the core of the algorithm: the particles communicate so that they acquire information on the regions explored by other particles. In fact, it has been claimed that the uniqueness of the PSO algorithm lies in the interactions of the particles [5]. As expected, the population topology deeply affects the balance between exploration and exploitation and the convergence speed and accuracy of the algorithm.

The population can be structured on any possible topology, from sparse to dense (or even fully connected) graphs, with different degrees of connectivity and clustering. The classical and most used population structures are the *lbest* (which connects the individuals to a local neighbourhood) and the *gbest* (in which each particle is connected to every other individual). These topologies are well-studied and the major conclusions are that *gbest* is fast but is frequently trapped in local optima, while *lbest* is slower but converges more often to the neighborhood of the global optima.

Since the first experiments on *lbest* and *gbest* structures, researchers have tried to design networks that hold the best traits given by each structure [9]. Some studies also try to understand what makes a good structure. For instance, Kennedy and Mendes [5] investigate several types of topologies and recommend the use of a lattice with von Neumann neighbourhood (which results in a connectivity degree between that of *lbest* and *gbest*).

Recently, dynamic structures have been introduced in PSO for improving the algorithm's adaptability to different fitness landscapes and overcome the rigidity of static structures, like [7], for instance. Fernandes et al. [1] try a different approach and propose a dynamic and partially connected von Neumann structure with Brownian motion. In this paper, we use the same model but a strategy for the conservation of function evaluations [8] is introduced in order to make the most of the underlying structure and reduce convergence speed. A formal description of the dynamic network is given here, opening the way for more sophisticated dynamics.

In the proposed topology, $n$ particles are placed in a 2-dimensional m-nodes grid where $m > n$. Every time-step, each individual checks its von Neumann neighborhood and, as in the standard PSO, updates its velocity and position using the information given by the neighbours. However, while the connectivity degree of the von Neumann topology is $k = 5$ the degree of the proposed topology is variable in the range $1 \leq k \leq 5$. Furthermore, the structure is dynamic: in each time-step, every particle updates its position on the grid (which is a different concept from the position of the particle on the fitness landscape) according to a pre-defined rule that selects the destination node. The movement rule, which is implemented locally and without any knowledge on the global state of the system, can be based on stigmergy [2] or Brownian motion.

As stated above, the connectivity degree $k$ of each particle in each time-step is variable and lies in the range $1 \leq k \leq 5$. Depending on the size of the grid, there may be particles with $k = 1$. These particles without neighbors (except the particle itself) do not learn from any local neighbourhood at that specific iteration. Therefore, it is expected that they continue to follow their previous trajectory in the fitness landscape. Taking into account these premises, the algorithm proposed in this study does not evaluate the position of the particles when $k = 1$. Regardless of the loss of informant intrinsic to a *conservation of evaluations* policy, we hypothesize that the strategy is particularly suited for the proposed dynamic topology (in which the particles are sometimes isolated from the flow of information) and the number of function evaluations required for meeting the stop criteria can be significantly reduced. Furthermore, it is the structure of the population and the position of the particles at a specific time-step that decides the application of the conservation rule and not any extra parameter or pre-defined decision rule.

A classical PSO experimental setup is used for the tests and the results demonstrate that the proposed algorithm consistently improves the speed of convergence of the standard von Neumann structure without degrading the quality of solutions. The experiments also demonstrate that the introduction of the conservation strategy reduces significantly the convergence speed without affecting the quality of the final solutions.

The remaining of the paper is organized as follows. Section 2 describes PSO and gives an overview on population structures for PSOs. Section 3 gives a formal description of the proposed structure. Section 4 describes the experiments and discusses the results and, finally, Sect. 5 concludes the paper and outlines future research.

## 2 Background Review

PSO is described by a simple set of equations that define the velocity and position of each particle. The position of the $i$th particle is given by $\vec{X}_i = (x_{i,1}, x_{i,2}, \ldots x_{1,D})$, where $D$ is the dimension of the search space. The velocity is given by $\vec{V}_i = (v_{i,1}, v_{i,2}, \ldots v_{1,D})$. The particles are evaluated with a fitness function $f(\vec{X}_i)$ and then their positions and velocities are updated by:

$$v_{i,d}(t) = v_{i,d}(t-1) + c_1 r_1 (p_{i,d} - x_{i,d}(t-1)) + c_2 r_2 \left(p_{g,d} - x_{i,d}(t-1)\right) \tag{1}$$

$$x_{i,d}(t) = x_{i,d}(t-1) + v_{i,d}(t) \tag{2}$$

were $p_i$ is the best solution found so far by particle $i$ and $p_g$ is the best solution found so far by the neighborhood. Parameters $r_1$ and $r_2$ are random numbers uniformly distributed in the range [0, 1] and $c_1$ and $c_2$ are acceleration coefficients that tune the relative influence of each term of the formula. The first term is known as the *cognitive* part, since it relies on the particle's own experience. The last term is the *social* part, since it describes the influence of the community in the trajectory of the particle.

In order to prevent particles from stepping out of the limits of the search space, the positions $x_{i,d}(t)$ are limited by constants that, in general, correspond to the domain of the problem: $x_{i,d}(t) \in [-Xmax, Xmax]$. Velocity may also be limited within a range in order to prevent the *explosion* of the velocity vector: $v_{i,d}(t) \in [-Vmax, Vmax]$.

For achieving a better balancing between local and global search, Shi an Eberhart [12] added the inertia weight $\omega$ as a multiplying factor of the first term of Eq. 1. This paper uses PSOs with inertia weight.

The neighbourhood of the particle defines the value of $p_g$ and is a key factor in the performance of PSO. Most of the PSOs use one of two simple sociometric principles for defining the neighbourhood network. One connects all the members of the swarm to one another, and it is called *gbest*, were $g$ stands for *global*. The degree of connectivity of *gbest* is $k = n$, where $n$ is the number of particles. Since all the particles are connected to every other and information spreads easily through the network, the *gbest* topology is known to converge fast but unreliably (it often converges to local optima).

The other standard configuration, called *lbest* (where $l$ stands for local), creates a neighbourhood that comprises the particle itself and its $k$ nearest neighbors. The most common *lbest* topology is the ring structure, in which the particles are arranged in a ring structure (resulting in a degree of connectivity $k = 3$, including the particle). The *lbest* converges slower than the *gbest* structure because information spreads slower through the network but for the same reason it is less prone to converge prematurely to local optima. In-between the ring structure with $k = 3$ and the *gbest* with $k = n$ there are several types of structure, each one with its advantages on a certain type of fitness landscapes. Choosing a proper structure depends on the target problem and also on the objectives or tolerance of the optimization process.

Kennedy and Mendes [5] published an exhaustive study on population structures for PSOs. They tested several types of structures, including the *lbest*, *gbest* and von Neumann configuration with radius 1 (also kown as *L*5 neighborhood). They also tested populations arranged in randomly generated graphs. The authors conclude that when the configurations are ranked by the performance the structures with $k = 5$ (like the *L*5) perform better, but when ranked according to the number of iterations needed to meet the criteria, configurations with higher degree of connectivity perform better. These results are consistent with the premise that low connectivity favors robustness, while higher connectivity favors convergence speed (at the expense of reliability). Amongst the large set of graphs tested in [5], the Von Neumann with radius 1 configuration performed more consistently and the authors recommend its use.

Alternative topologies that combine standard structures' characteristics or introduce some kind of dynamics in the connections have been also proposed. Parsopoulos and Vrahatis [9] describe the unified PSO (UPSO), which combines the *gbest* and *lbest* configurations. Equation 1 is modified in order to include a term with $p_g$ and a term with $p_i$ and a parameter balances the weight of each term. The authors argue that the proposed scheme exploits the good properties of *gbest* and *lbest*.

Peram et al. [10] proposed the fitness-distance-ratio-based PSO (FDR-PSO), which defines the neighbourhood of a particle as its $k$ closest particles in the population (measured by the Euclidean distance). A selective scheme is also included: the particle selects nearby particles that have also visited a position of higher fitness. The algorithm is compared to a standard PSO and the authors claim that FDR-PSO performs better on several test functions. However, the FDR-PSO is compared only to a *gbest* configuration, which is known to converge frequently to local optima in the majority of the functions of the test set.

More recently, a comprehensive-learning PSO (CLPSO) was proposed [7]. Its learning strategy abandons the global best information and introduces a complex and dynamic scheme that uses all other particles' past best information. CLPSO can significantly improve the performance of the original PSO on multimodal problems. Finally, Hseigh et al. [3] use a PSO with varying swarm size and solution-sharing that, like in [7], uses the past best information from every particle.

A different approach is given in 1. The authors describe a structure that is based on a grid of $m$ nodes (with $m > n$) on which the particles move and interact. In this structure, a particle, at a given time-step, may have no neighbours except itself. The isolated particles will continue to follow its previous trajectory, based on their current information, until they find another particle in the neighbourhood. Therefore, we intend to investigate if the loss of information caused by not evaluating these particles is overcome by the payoff in the convergence speed.

Common ways of addressing the computational cost of evaluating solutions in hard real-world problems are function approximation [6], fitness inheritance [11] and conservation of evaluations [8]. Due to the underlying structure of the proposed algorithm, we have tested a conservation policy similar to the GREEN-PSO proposed by Majercik [8]. However, in our algorithm the decision on evaluating or not is defined by the position of the particle in the grid (isolated particles are not evaluated) while in the GREEN-PSO the decision is probabilistic and the likelihood of conserving a solution is controlled by a parameter.

The following section gives a formal description of the proposed network and presents the transition rules that define the model for dynamic population structures.

## 3 Partially Connected Structures

Let us consider a rectangular grid $G$ of size $q \times s \geq \mu$, where $\mu$ is the size of the population of any population-based metaheuristics or model. Each node $G_{uv}$ of the grid is a tuple $\langle \eta_{uv}, \zeta_{u,v} \rangle$, where $\eta_{uv} \in \{1, \ldots, \mu\} \cup \{\bullet\}$ and $\zeta_{uv} \in (D \times \mathbb{N}) \cup \{\bullet\}$ for some domain $D$. The value $\eta_{uv}$ indicates the index of the individual that occupies the position $\langle u, v \rangle$ in the grid. If $\eta_{uv} = \bullet$ then the corresponding position is empty. However, that same position may still have information, namely a mark (or clue) $\zeta_{uv}$. If $\zeta_{uv} = \bullet$ then the position is empty and unmarked. Please note that when $q \times s = \mu$, the topology is a static 2-dimensional lattice and when $q \times s = \mu$ and $q = s$ the topology is the standard square grid graph.

In the case of a PSO, the marks are placed by particles that occupied that position in the past and they consist of information about those particles, like their fitness $\zeta^f_{uv}$ or position in the fitness landscape, as well as a time stamp $\zeta^t_{uv}$ that indicates the iteration in which the mark was placed. The marks have a lifespan of $K$ iterations, after which they are deleted.

Initially, $G_{uv} = (\bullet, \bullet)$ for all $\langle u, v \rangle$. Then, the particles are placed randomly on the grid (only one particle per node). Afterwards, all particles are subject to a movement phase (or grid position update), followed by a PSO phase. The process (position update and PSO phase) repeats until a stop criterion is met.

The PSO phase is the standard iteration of a PSO, comprising position and velocity update. The only difference to a static structure is that in this case a particle may find empty nodes in its neighbourhood.

In the position update phase, each individual moves to an adjacent empty node. Adjacency is defined by the Moore neighborhood of radius $r$, so an individual $i$ at $\rho_g(i) = \langle u, v \rangle$ can move to an empty node $\langle u', v' \rangle$ for which $L_\infty(\langle u, v \rangle \langle u', v' \rangle) \leq r$. If empty positions are unavailable, the individual stays in the same node. Otherwise, it picks a neighboring empty node according to the marks on them. If there are no marks, the destination is chosen randomly amongst the free nodes.

With this framework, there are two possibilities for the position update phase: stimergic, whereby the individual looks for a mark that is similar to itself; and Brownian, whereby the individual selects an empty neighbor regardless of the marks. For the first option, let $\mathbb{N}\langle u, v \rangle = \{\langle u^{(1)}, v^{(1)} \rangle, \ldots \langle u^w, v^w \rangle\}$ be the collection of empty neighboring nodes and let $i$ be the individual to move. Then, the individual attempts to move to a node whose mark is as close as possible to its own corresponding trait (fitness or position in the fitness landscape, for instance) or to an adjacent cell picked at random if there are no marks in the neighborhood. In the alternative Brownian policy, the individual moves to an adjacent empty position picked at random. In either case, the process is repeated for the whole population.

For this paper, the investigation is restricted to the Brownian structure. The algorithm is referred in the remaining of the paper has PSO-B, followed by the grid size $q \times s$. An extension of the PSO-B is constructed by introducing a conservation of function evaluations (*cfe*) strategy. If at a given time-step a particle has no neighbors, then the particle is updated but its position is not evaluated. This version of the algorithm is referred to as PSO-B*cfe*. The following section describes the results attained by the PSOs with dynamic structure and Brownian movement, with and without conservation of function evaluations and compares them to the standard topology.

## 4 Experiments and Results

An experimental setup was constructed with eight benchmark unimodal and multimodal functions that are commonly used for investigating the performance of PSO. The functions are described in Table 1. The dimension of the search space is set to $D = 30$ (except Schaffer, with $D = 2$). In order to obtain a square grid graph for the standard *von Neumann* topology, the population size $n$ is set to 49 (which is

**Table 1** Benchmarks for the experiments

| Function | Mathematical representation | Range of search/Range of initialization | Stop criteria |
|---|---|---|---|
| *Sphere $f_1$* | $f_1(\vec{x}) = \sum_{i=}^{D} x_i^2$ | $(-100, 100)^D$<br>$(50, 100)^D$ | 0.01 |
| *Rosenbrock $f_2$* | $f_2(\vec{x}) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$ | $(-100, 100)^D$<br>$(15, 30)^D$ | 100 |
| *Rastrigin $f_3$* | $f_3(\vec{x}) = \sum_{i=1}^{D-1} (x_i^2 - 10\cos(2\pi x_i) + 10)$ | $(-10, 10)^D$<br>$(2.56, 5.12)^D$ | 100 |
| *Griewank $f_4$* | $f_4(\vec{x}) = 1 + \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$ | $(-600, 600)^D$<br>$(300, 600)^D$ | 0.05 |
| *Schaffer $f_5$* | $f_5(\vec{x}) = 0.5 + \frac{\left(\sin\sqrt{x^2+y^2}\right)^2 - 0.5}{(1.0 + 0.001(x^2+y^2))^2}$ | $(-100, 100)^2$<br>$(15, 30)^2$ | 0.00001 |
| *Weierstrass $f_6$* | $f_6(\vec{x}) = \sum_{i=1}^{D} \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (x_i + 0.5))]\right)$<br>$- D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k \cdot 0.5)],$<br>$a = 0.5, b = 3, kmax = 20$ | $(-0.5, 0.5)^D$<br>$(0.5, 0.2)^D$ | 0.01 |
| *Ackley $f_7$* | $f_7(\vec{x}) = -20 exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)$<br>$- exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20$<br>$+ e$ | $(-32.768, 32.768)^D$<br>$(2.56, 5.12)^D$ | 0.01 |
| *Schwefel $f_8$* | $f_7(\vec{x}) = 418.9829 \times D - \sum_{i=1}^{D} x_i sin\left(\left\vert x_i^{\frac{1}{2}}\right.\right)$ | $(-500, 500)^D$<br>$(-500, 500)^D$ | 3000 |

Dynamic range, initialization range and stop criteria within the typical range of PSO's swarm size). The acceleration coefficients were set to 1.494 and the inertia weight is 0.729, as in [13]. *Xmax* is defined as usual by the domain's upper limit and $Vmax = Xmax$. A total of 50 runs for each experiment are conducted. *Asymmetrical initialization* is used (the initialization range for each function is given in Table 1).

Two experiments were conducted. Firstly, the algorithms were run for a limited amount of function evaluations (147000 for $f_1$ and $f_5$, 49000 for the remaining) and the fitness of the best solution found was averaged over 50 runs. In the second experiment the algorithms were run for 980000 evaluations (corresponding to 20000 iterations of standard PSO with $n = 49$) or until reaching a stop criterion. For each function and each algorithm, the evaluations required to meet the criterion was recorded and averaged over the 50 runs. A success measure is defined as the number of runs in which an algorithm attains the fitness value established as the stop criterion.

Tables 1 and 2 compare PSO-B with the standard PSO (with von Neumann topology): Table 1 gives the averaged best fitness found by the swarms while Table 2 gives, for each algorithm and each function, the averaged number of iterations required to meet the criterion, and the number of runs in which the criterion was met.

The best fitness values are similar in both configurations. In fact, the differences are not statistical significant except for function $f_1$, for which PSO-Bcfe significantly better than PSO. (For the statistical tests comparing two algorithms, non-parametric Kolmogorov-Smirnov tests (with 0.05 level of significance) have been used.) As for the convergence speed, PSO-B is faster in every test function. The results are significantly different in $f_1$, $f_2$, $f_3$, $f_5$, $f_6$ and $f_8$. PSO-B and the standard PSO attain similar fitness values, but PSO-B is faster.

The main hypothesis of this paper is that a conservation of evaluations strategy further improves the convergence speed of the dynamic topology. Moreover, we also expect that PSO-B*cfe* performance is less affected when the size of the grid is increased. Large grid sizes result in large rates of isolated particles, deprived from social information, which reduces the convergence speed of the algorithm. By not evaluating these particles, the computational effort can be significantly reduced, hopefully without degrading the overall performance. In order to investigate these hypotheses, we have compared PSO-B and PSO-B*cfe*, while varying the size of the grid (Table 3).

Table 4 shows the average fitness values attained by PSO-B and PSO-B*cfe* with different grid sizes. Table 5 displays the average number of function evaluations required to meet the stop criteria as well as the number of successful runs. The performance according to the fitness values is very similar, with no significant differences between the algorithm in every function except $f_1$ (in which PSO-B*cfe* is significantly better). When considering the number of function evaluations (i.e., the convergence speed), PSO-Bcfe is significantly better or statistically equivalent in every function.

The results confirm that PSO-B*cfe* is able to improve the convergence speed of PSO-B without degrading the accuracy of the solutions. The loss of information

**Table 2** Standard PSO with von Neumann topology and Brownian PSO with 10 × 10 grid

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|---|---|---|---|---|---|---|---|---|
| PSO | 4.26e − 36 | 1.39e + 01 | 6.40e + 01 | **5.61e − 03** | 0.00e + 00 | **4.43e − 02** | **1.12e − 15** | 3.07e + 03 |
| | ±1.32e − 35 | ±2.39e + 01 | ±1.59e + 01 | **±8.78e − 03** | ±0.00e + 00 | ±1.69e − 01 | ±2.24e − 16 | ±4.92e + 02 |
| PSO-B | **8.74e − 38** | **1.25e + 01** | **6.23e + 01** | 7.73e − 03 | **0.00e + 00** | 4.61e − 02 | **1.12e − 15** | **2.80e + 03** |
| | **±1.29e − 37** | **±2.07e + 01** | **±1.88e + 01** | ±9.50e − 03 | **±0.00E + 00** | ±2.50e − 01 | **±2.24e − 16** | **±6.85e + 02** |

Best fitness values averaged over 50 runs

**Table 3** Standard PSO with von Neumann topology and Brownian PSO (10 × 10)

| | *f₁* | *f₂* | *f₃* | *f₄* | *f₅* | *f₆* | *f₇* | *f₈* |
|---|---|---|---|---|---|---|---|---|
| PSO | 23530.78 | 72707.18 | 18424.00 | 22015.70 | 17622.36 | 41677.84 | 24420.62 | 21384.05 |
| | ±954.74<br>(50) | ±92916.33<br>(50) | ±11082.75<br>(49) | ±1304.60<br>(50) | ±16056.68<br>(50) | ±1360.37<br>(44) | ±958.81<br>(50) | ±20450.21<br>(22) |
| PSO-B | **22700.72** | **65769.76** | **15114.46** | **21574.70** | **10741.78** | **38976.00** | **24410.86** | **12943.90** |
| | **±906.39**<br>**(50)** | **±70232.15**<br>**(50)** | **±3939.74**<br>**(48)** | **±1107.56**<br>**(50)** | **±10658.16**<br>**(50)** | **±1654.12**<br>**(42)** | **±1533.28**<br>**(50)** | **±3422.18**<br>**(31)** |

Function evaluations averaged over 50 runs and success rates

**Table 4** PSO-B and PSO-B*cfe*

| | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
|---|---|---|---|---|---|---|---|---|
| PSO-B<br>8 × 8 | 2.71e − 40 | 7.98e + 00 | 6.57e + 01 | 7.43e − 03 | 1.17e − 03 | 1.55e − 01 | 1.17e − 15 | 2.85e + 03 |
| | ±4.21e − 40 | ±1.11e + 01 | ±1.78e + 01 | ±9.02e − 03 | ±3.19e − 03 | ±4.04e − 01 | ±2.15e − 16 | ±5.05e + 02 |
| PSO-B*cfe*<br>8 × 8 | 3.20e − 40 | 1.12e + 01 | 6.29e + 01 | 7.44e − 03 | 1.94e − 04 | 3.59e − 02 | 1.20e − 15 | 2.78e + 03 |
| | ±8.44e − 40 | ±2.02e + 01 | ±1.35e + 01 | ±7.96e − 03 | ±1.37e − 03 | ±2.15e − 01 | ±2.4 − 16 | ±5.48e + 02 |
| PSO-B<br>10 × 10 | 8.74e − 38 | 1.25e + 01 | 6.23e + 01 | 7.73e − 03 | **0.00e + 00** | 4.61e − 02 | **1.12e − 15** | 2.80e + 03 |
| | ±1.29e − 37 | ±2.07e + 01 | ±1.88e + 01 | ±9.50e − 03 | **±0.00e + 00** | ±2.50e − 01 | **±2.24e − 16** | ±6.85e + 02 |
| PSO-B*cfe*<br>10 × 10 | 2.43e − 39 | **1.09e + 01** | 6.16e + 01 | 5.61e − 03 | **0.00e + 00** | 1.15e − 01 | 1.19e − 15 | 2.91e + 03 |
| | ±4.38e − 39 | **±1.67e + 01** | ±1.60e + 01 | ±7.47e − 03 | **±0.00e + 00** | ±3.30e − 01 | ±2.09e − 16 | ±5.37e + 02 |
| PSO-B<br>15 × 15 | 3.47e − 33 | 1.53e + 01 | **6.05e + 01** | **3.79e − 03** | 3.89e − 04 | 3.65e − 02 | 1.14e − 15 | **2.72e + 03** |
| | ±4.31e − 33 | ±2.65e + 01 | **±1.45e + 01** | **±6.22e − 03** | ±1.92e − 03 | ±2.15e − 01 | ±2.23e − 16 | **±5.07e + 02** |
| PSO-B*cfe*<br>15 × 15 | **2.47e − 45** | 1.31 e + 01 | 6.62e + 01 | 6.45e − 03 | **0.00e + 00** | **3.59e − 02** | 1.13e − 15 | **2.72e + 03** |
| | **±4.21e − 45** | ±2.42e + 01 | ±2.05e + 01 | ±9.90e − 03 | **±0.00E + 00** | **±2.15e − 01** | ±2.24e − 16 | **±5.81e + 02** |

Best fitness values averaged over 50 runs

**Table 5** PSO-B and PSO-B*cfe*

| | *f₁* | *f₂* | *f₃* | *f₄* | *f₅* | *f₆* | *f₇* | *f₈* |
|---|---|---|---|---|---|---|---|---|
| PSO-B<br>8 × 8 | 21070.98 | 56472.50 | 13224.90 | 19959.66 | 14369.25 | 37385.64 | 22112.72 | **10756.34** |
| | ±1023.60 (50) | ±49342.60 (50) | ±3894.30 (48) | ±1408.36 (50) | ±20599.03 (46) | ±1864.74 (36) | ±1172.33 (50) | **±2866.56 (29)** |
| PSO-B*cfe*<br>8 × 8 | 21157.7 | 96148.42 | 13427.12 | 19968.40 | 10275.69 | 33519.66 | 21975.32 | 11477.00 |
| | ±1092.81 (50) | ±112675.6 (50) | ±2259.73 (50) | ±1103.20 (50) | ±6952.548 (49) | ±1522.85 (33) | ±1079.08 (50) | ±3318.38 (33) |
| PSO-B<br>10 × 10 | 22700.72 | 65769.76 | 15114.46 | 21574.70 | 10741.78 | 38976.00 | 24310.86 | 12943.90 |
| | ±906.39 (50) | ±70232.15 (50) | ±3939.74 (48) | ±1107.56 (50) | ±10658.16 (50) | ±1654.12 (42) | ±1533.28 (50) | ±3422.18 (31) |
| PSO-B*cfe*<br>10 × 10 | 21796.04 | **57704.16** | 13953.04 | 20430.34 | 11817.06 | 39002.15 | 23159.40 | 12191.81 |
| | ±832.54 (50) | **±71260.77 (50)** | ±3341.41 (49) | ±1176.32 (50) | ±15647.63 (50) | ±2753.01 (40) | ±1105.97 (50) | ±2252.50 (27) |
| PSO-B<br>15 × 15 | 26122.88 | 74321.24 | 20408.50 | 24626.42 | 11830.83 | 45721.17 | 27151.88 | 20961.41 |
| | ±950.08 (50) | ±83535.59 (50) | ±3692.10 (50) | ±1406.53 (50) | ±11576.37 (48) | ±1471.31 (37) | ±966.74 (50) | ±11807.10 (37) |
| PSO-B*cfe*<br>15 × 15 | **19600.76** | 77348.06 | **16713.59** | **18734.94** | **10890.55** | **33519.66** | **20345.96** | 13614.55 |
| | **±730.62 (50)** | ±91374.00 (50) | **±4387.78 (46)** | **±1029.26 (50)** | **±11624.59 (47)** | **±1522.85 (38)** | **±795.368 (50)** | ±5513.17 (38) |

Best fitness values averaged over 50 runs

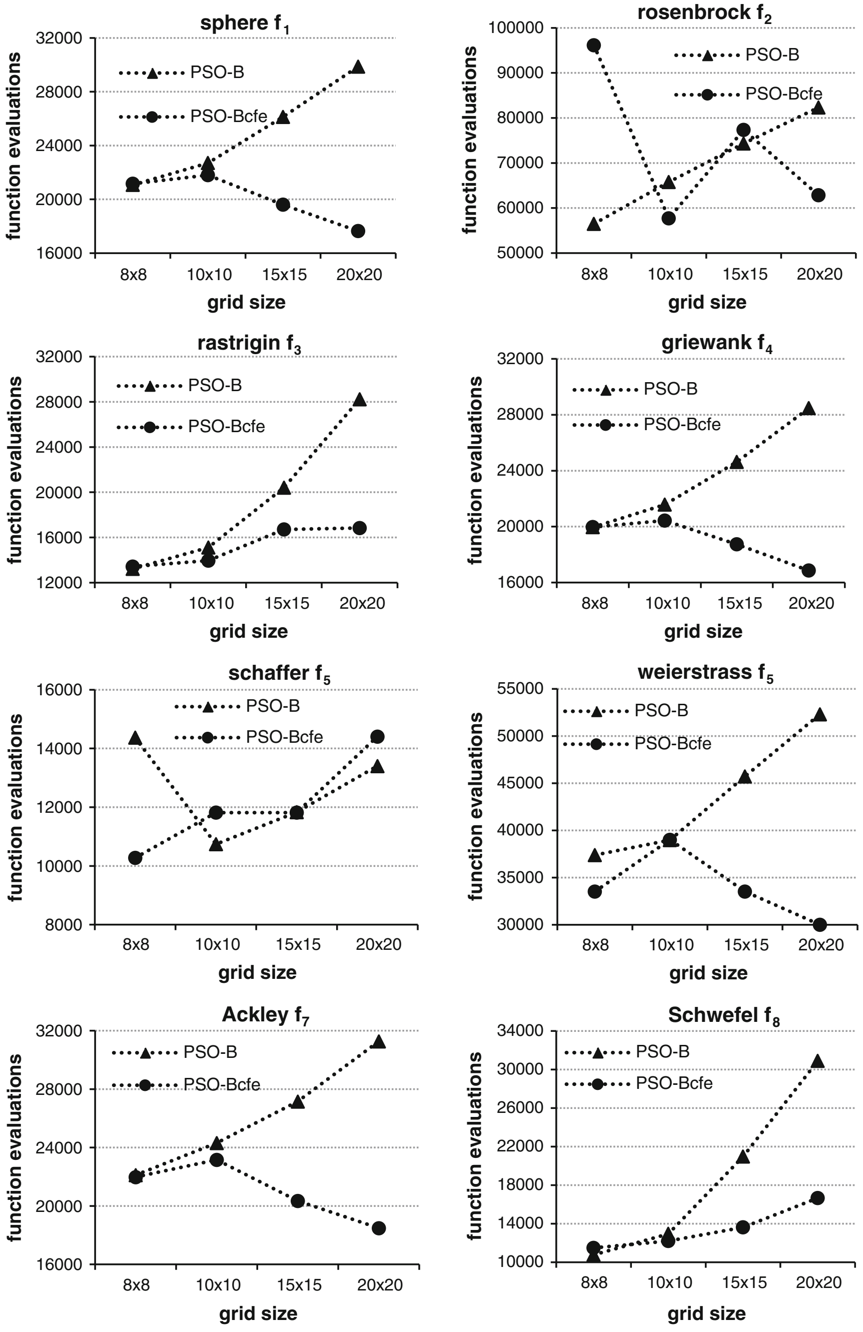

**Fig. 1** PSO-B and PSO-B*cfe*. Function evaluations required to meet stop criteria when using grids with different sizes

that results from conserving evaluations is clearly overcome by the benefits of reducing the computational cost per iteration.

In the case of $f_1$, PSO-B*cfe* also significantly improves the quality of the solutions, namely with larger grids. The proposed scheme seems to be particularly efficient in unimodal landscapes, but further tests are required in order to confirm this hypothesis and understand what mechanisms make PSO-B*cfe* so efficient in finding more precise solutions for the sphere function.

The differences in the convergence speed of the algorithm are particularly noticeable when the grid is larger. While PSO-B's speed tends to decrease when the grid size increases, the behavior of PSO-B*cfe*, is much more stable, and in some functions it is even faster when the grid is expanded.

Figure 1 graphically depicts the above referred observations. When the grid size grows from 8 × 8 to 20 × 20, PSO-B's convergence speed degrades consistently, except in function $f_5$, where the behavior is more irregular. PSO-Bcfe, on the other hand, is sometimes faster with larger grids. When its convergence speed decreases with size ($f_8$, for instance), it scales better than PSO-B.

## 5 Conclusions

This paper proposes a general scheme for structuring dynamic populations for the Particle Swarm Optimization (PSO) algorithm. The particles are placed on a grid of nodes where the number of nodes is larger than the swarm size. The particles move on the grid according to simple rules and the network of information is defined by the particle's position on the grid and its neighborhood (*von Neumann* vicinity is considered here). If isolated (i.e., no neighbors except itself), the particle is updated but its position is not evaluated. This strategy may result in some loss of information, but the results show that the payoff in convergence speed overcomes the loss of information: the convergence speed is increased in the entire test set, while the accuracy of the algorithm (i.e., the averaged final fitness) is not degraded by the conservation of evaluations strategy.

The proposed algorithm is tested with a Brownian motion rule and compared to the standard static topology. The conservation of evaluations strategy results in a more stable performance when varying the grid size. Removing the strategy from the proposed dynamic structure results in a drop of the convergence speed when the size of the grid increases in relation to the swarm size.

The present study is restricted to dynamic structures based on particles with Brownian motion. Future research will be focused on dynamic structures with stigmergic behavior based on the fitness and position of the particles.

**Acknowledgments** The first author wishes to thank FCT, *Ministério da Ciência e Tecnologia*, his Research Fellowship SFRH/BPD/66876/2009. The work was supported by FCT PROJECT [PEst-OE/EEI/LA0009/2013], Spanish Ministry of Science and Innovation projects TIN2011-28627-C04-02 and TIN2011-28627-C04-01, Andalusian Regional Government P08-TIC-03903 and P10-TIC-6083, CEI-BioTIC UGR project CEI2013-P-14, and UL-EvoPerf project.

## References

1. Fernandes, C.M., Laredo, J.L.J., Merelo, J.J., Cotta, C., Nogueras, R., Rosa, A.C.: Performance and scalability of particle Swarms with dynamic and partially connected grid topologies. In: Proceedings of the 5th International Joint Conference on Computational Intelligence IJCCI 2013, pp. 47–55 (2013)
2. Grassé, P.-P.: La reconstrucion du nid et les coordinations interindividuelles chez bellicositermes et cubitermes sp. La théorie de la stigmergie: Essai d'interpretation du comportement des termites constructeurs, Insectes Sociaux **6**, 41–80 (1959)
3. Hseigh, S.-T., Sun, T.-Y, Liu, C.-C., Tsai, S.-J.: Efficient population utilization strategy for particle swarm optimizers. IEEE Trans. Syst., Man Cybern. Part B **39**(2), 444–456 (2009)
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. Proc. IEEE Int. Conf. Neural Netw. **4**, 1942–1948 (1995)
5. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Proceedings of the IEEE World Congress on Evolutionary Computation, pp. 1671–1676 (2002)
6. Landa-Becerra, R., Santana-Quintero, L.V., Coello Coello, C.A.: Knowledge incorporation in multi-objective evolutionary algorithms. In: Multi-Objective Evolutionary Algorithms for Knowledge Discovery from Databases, pp. 23–46 (2008)
7. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput. **10**(3), 281–296 (2006)
8. Majercik, S.: GREEN-PSO: conserving function evaluations in particle swarm optimization. In: Proceedings of the IJCCI 2013—International Joint Conference on Computational Intelligence, pp. 160–167 (2013)
9. Parsopoulos, K.E., Vrahatis, M.N.: UPSO: a unified particle swarm optimization scheme. In: Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2004), pp. 868–887 (2004)
10. Peram, T., Veeramachaneni, K., Mohan, C.K.: Fitness-distance-ratio based particle swarm optimization. In: Proceedings of the Swarm Intelligence Symposium SIS'03, pp. 174–181 (2003)
11. Reyes-Sierra, M., Coello Coello, C.A.: A study of techniques to improve the efficiency of a multiobjective particle swarm optimizer. In: Studies in Computational Intelligence (51), Evolutionary Computation in Dynamic and Uncertain Environments, pp. 269–296 (2007)
12. Shi, Y., Eberhart, R.C.: A Modified Particle Swarm Optimizer. In: Proceedings of IEEE 1998 International Conference on Evolutionary Computation, pp. 69–73. IEEE Press (1998)
13. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf. Proc. Lett. **85**, 317–325 (2003)