Multi-Ring Particle Swarm Optimization

Carmelo J. A. Bastos-Filho, Danilo F. Carvalho, Marcel P. Caraciolo, Péricles B. C. Miranda and Elliackin M. N. Figueiredo
Department of Computing and Systems, University of Pernambuco
Brazil

1. Introduction

Many real world problems are quite expensive to be solved by using exhaustive computation. Because of this, many metaheuristics have been proposed recently to deal with optimization and search problems. Although many approaches have been proposed based on local search, the current research in Evolutionary Computation (EC) is mainly focused on algorithms that are based on populations and inspired in nature (Schwefel, 1981; Bäck, 1996; Fogel, 1996; Beyer, 2001; Beyer, 2002).

In this context, Swarm intelligence techniques, mainly Particle Swarm Optimization (PSO), have been widely applied to solve many different types of optimization problems. PSO was proposed by Kennedy and Eberhart (Kennedy & Eberhart, 1995) and the main idea is to explore the intelligence originated by the social knowledge disseminated by the agents during the search process. Due to its great performance in difficult optimization tasks and its simplicity, PSO became a popular, easy-to-use and extendable technique (Abido, 2001; Eberhart & Shi, 2001; Parsopoulos & Vrahatis, 2001; Agrafiotis & Cedeno, 2002; Coello Coello & Lechuga, 2002; Parsopoulos et al., 2002; Shi & Krohling, 2002; Praveen et al., 2007; Santana-Quintero ET AL., 2008).

By applying PSO to problems where the feasible solutions are too much difficult to find, more sophisticated approaches are required, mainly for hyper dimensional spaces. Many variations on the basic PSO form have been explored, most of them proposing different velocity update equations (Shi & Eberhart, 1998; Clerc & Kennedy, 2002; Kennedy, 2005; Jiao, et al., 2008). Other approaches attempt to change the structure of the swarm (Kennedy & Mendes, 2002; Suganthan, 1999) and its communication schema (Engelbrecht, 2005; Sutton et al., 2006).

To disseminate information within the swarm is the key of any swarm intelligence based algorithm. PSO, like others swarm algorithms, makes use of its own information exchange methods to distribute the best positions found during the algorithm execution (Engelbrecht, 2005). The way used by the swarm to distribute this information lies in the social structure formed by the particles. Variations in this structure can improve the algorithm performance.

By using local communication, each particle of the swarm has its own neighborhood. Many papers (Engelbrecht, 2005; Carvalho & Bastos-Filho, 2008; Carvalho & Bastos-Filho, 2009) have shown that it improves the exploration capacity of the entire swarm but decreases the convergence speed of the algorithm. It means that the algorithm obtains better solutions, but it needs more iterations to reach the convergence.

Source: Evolutionary Computation, Book edited by: Wellington Pinheiro dos Santos,
ISBN 978-953-307-008-7, pp. 572, October 2009, I-Tech, Vienna, Austria

Other important and already known PSO algorithm variation is the Cooperative Particle Optimizer (CPSO) (van den Bergh & Engelbrecht, 2001; van den Bergh & Engelbrecht, 2004) which is based on the cooperative GA proposed by Potter and De Jong (Potter & De Jong, 1994). This approach proposes a performance improvement of the PSO algorithm by splitting the solution vector into different components that will be optimized by different sub-swarms cooperatively. CPSO inserts another parameter called *split factor* which can be interpreted as the number of sub-swarms that compose the entire swarm.

By using the ring structure as an inspiration, a novel PSO topology called Multi-ring PSO (MRPSO) based on coupling different rings was recently proposed (Bastos-Filho *et al.* 2008). This novel approach makes use of a diversity mechanism provided by ring rotations in order to improve the exploration of the search space. The structure of the proposed topology is formed by ring layers connected by axis that cross the layers trough the particles. Each single particle, which belongs to a specific layer, is automatically connected to other particles in neighbour ring layers. These connections establish the communication between neighbour particles which is a peculiar neighbourhood scenario based on local communication. The ring layers increase the capacity to avoid stagnation states making use of a process and an attribute called rotation skill and trigger criterion, respectively, also described in the chapter. Furthermore, we included a novel concept proposed by Cai *et al.* (Cai *et al.* 2008) that assigns a grade to the particles based on their fitness. This information is used to switch the neighbourhood inside the rings.

The rest of this paper is organized as follows. Section 2 shows a brief review about PSO. It also describes some well known topologies. In Section 3 the proposed Multi-ring topology and its features are described. Section 4 is devoted to the Dispersed based Multi-ring approach. The simulation setup used by the experiments and the chosen benchmark functions are presented in Section 6. In section 7 the analysis of Multi-ring and Multi-ring Dispersed are shown. The results of a comparison with some well-known topologies are presented too.

2. Background

2.1 Particle Swarm Optimization - PSO

The basic form of the PSO algorithm is composed by a group particles, where each particle represents a possible solution to the problem. Each particle maps the attributes of a solution, *i. e.*, the position of one particle represents exactly the values of the attributes of a solution. Besides, particles move inside the search space reaching new positions. After some iterations, particles tend to a global optimum based on two basic components: the cognitive component and the social component. The first one considers the best position already found by the particle itself (***pbest***). The second one use the best position attained by all the particles of the swarm (***gbest***) in order to guide the swarm to a common spot in the search space. Based on these two information components, each particle updates its own velocity. The velocity is a vector that directs the particle to the next position.

As the original PSO is based on the global knowledge of the swarm, it has a low capacity for local search. Thus, other approaches were proposed in order to avoid this disadvantage. Shi and Eberhart (Shi & Eberhart, 1998) proposed a velocity update equation that uses an inertia factor w (see equation 1). This factor balances the exploration-exploitation trade-off by controlling the contribution of the previous velocity in the evaluation of the current velocity of each particle. For each iteration of the PSO algorithm the velocity $\vec{v}_i(t)$ contribution is decreased. Thus, the exploitation level of the swarm is increased.

$$\vec{v}_i(t+1) = \omega\vec{v}_i(t) + c_1 r_1(\vec{p}_i(t) - \vec{x}_i(t)) + c_2 r_2(\vec{p}_g(t) - \vec{x}_i(t)), \quad (1)$$

where *w* is the inertia factor, $\vec{x}_i(t)$ is the current position of particle *i* in time step *t*, c_1 and c_2 are positive acceleration constants, and *r1* and *r2* are two random numbers generated by a uniform distribution in the interval [0,1].

Recently, Jiao *et al*. (Jiao *et al*., 2008) proposed a dynamic inertia weight for the particle swarm optimization algorithm. They showed that it can improve the quality of the solutions obtained by the algorithm.

Another popular approach for the PSO velocity update equation involves a constriction factor which implies a velocity value reduction at each time step (Clerc & Kennedy, 2002). Under some peculiar conditions, it can guarantee both convergence and a high degree of exploration.

Both inertia and constriction approaches balance exploration and exploitation abilities and improve the convergence speed and the quality of the solutions found by the algorithm (Bratton & Kennedy, 2007).

2.2 Topologies

The communication between particles inside a swarm is one of the most important mechanisms of PSO. This mechanism distributes information already achieved by the

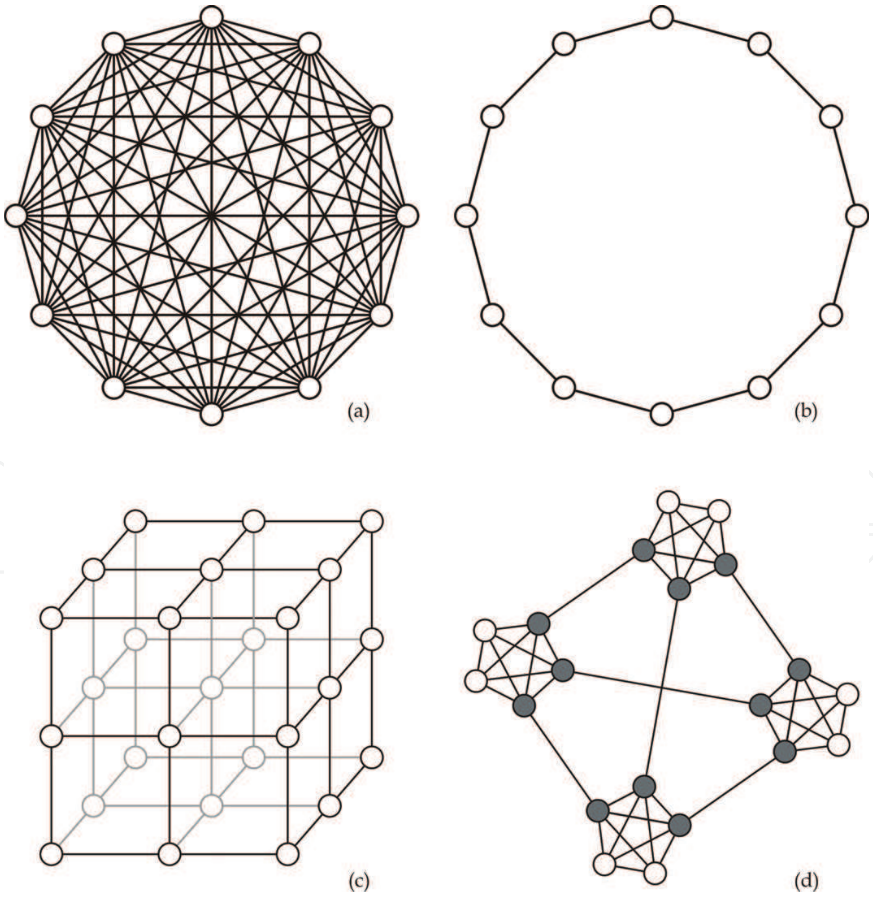


Fig. 1. Commonly used particle topologies: (a) Star topology used in ***gbest*** , (b) Ring topology used in ***lbest***, (c) Von Neumann topology and (d) Four clusters topology.

swarm through all particles, updating the swarm status. Global and local communication structures are the two main structures used for particles' communication. Considering that particles share information acquired through a structured swarm, this model of communication can be seen as a topology. Some of the well-known topologies are depicted in Figure 1.

Each topology shown in Figure 1 has its own way to spread information through the swarm (Kennedy, 1999; Kennedy & Mendes, 2002).

Star and *ring* topologies are shown in Figure 1(a) and Figure 1(b), respectively. *Star* topology is composed by a fully-connected structure (***gbest***) that allows each particle of the swarm to share information globally. In a global neighbourhood, information is instantaneously disseminated to all particles, quickly attracting the entire swarm to the same swarm region. To avoid local minima, *ring* topology (Figure 1(b)) make use of local neighbourhoods (***lbest***), *i. e.*, particles just communicate with its immediate neighbours. Therefore, the swarm needs a great number of iterations to converge, but higher quality solutions can be found.

In Figure 1(c), a grid of particles forming a peculiar social structure is shown. This structure is very useful for many optimization problems (Kennedy & Mendes, 2002; Peer et al., 2003) and is known as *Von Neumann* topology.

Based on clusters of particles, the *four clusters* topology is shown in Figure 1(d). Each cluster communicates with the others by connections previous defined between the particles (Mendes et al., 2003; Engelbrecht, 2005). The number of connections originated from a cluster is equal to the number of neighbour clusters.

3. Multi-ring topology

3.1 Multi-ring concepts

Multi-Ring is a novel PSO topology that uses the *ring* structure as an inspiration. By using local communication, each particle of the swarm has its own neighbourhood. Many papers have shown that it improves the explorarion capacity of the entire swarm but decreases the convergence time of the algorithm (Engelbrecht, 2005; Carvalho & Bastos-Filho, 2008; Carvalho & Bastos-Filho, 2009). A topology based on coupling different rings with dynamic commutation of the neighbourhood is proposed, which aims to obtain better solutions as the benefit of the local communication. At the same time, we expect to reduce the number of iterations to reach a convergence as the benefit of the dynamic neighbourhood mechanism.

The proposed topology is formed by particles arranged in multiple rings layers, where each particle of the ring can communicate with some particles of neighbour rings. In order to

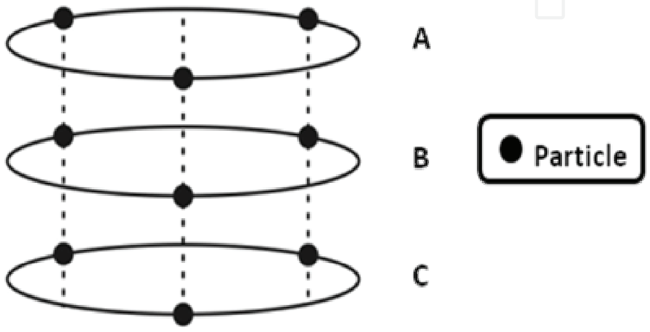


Fig. 2. Multi-Ring Topology.

couple different ring layers, the number of particles in each ring must be the same. Therefore, each single particle, which belongs to a specific layer, is automatically connected to other particles from neighbour ring layers, as shown in Figure 2. The structure in Figure 2 illustrates an example of the multi-ring topology with 3 rings (A, B and C) of 3 particles.

3.2 Communication between different rings

The particles from different ring layers communicate with each other by using a peculiar neighbourhood scenario. The Figure 3 can be used to illustrate this scenario. Considering a ring layer $rl_{(k)}$, each particle $rl_{(k)(i)}$ is directly connected with the particles $rl_{(k-1)(i)}$ and $rl_{(k+1)(i)}$ of the next and previous ring layers. During the algorithm execution, one particle $rl_{(k)(i)}$ will exchange information with a neighbourhood composed by $\{ rl_{(k)(i-1)}, rl_{(k)(i+1)}, rl_{(k-1)(i)}, rl_{(k+1)(i)} \}$.

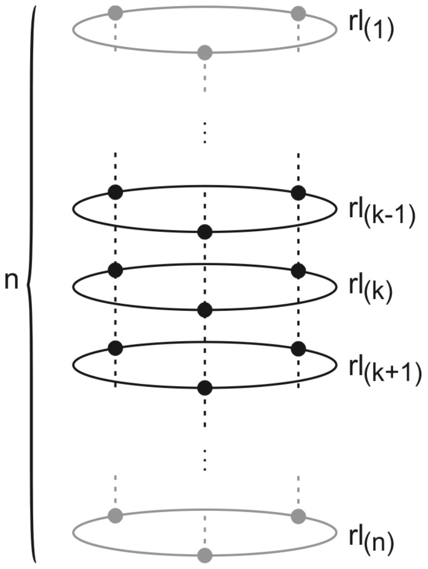


Fig. 3. Ring layers structure.

One special observation must be considered for the first and last ring layers of the topology. The communication between the particles located at those two rings is partially limited, since both do not have a complete ring neighbourhood, it means that they only have a unique ring layer directly connected to them. Therefore, the information exchange for one particle belonging to a ring $rl_{(1)}$ is limited by the set $\{ rl_{(1)(i-1)}, rl_{(1)(i+1)}, rl_{(2)(i)} \}$. The same restriction must be considered for the ring $rl_{(n)}$, which the neighbourhood is composed by the set $\{ rl_{(n)(i-1)}, rl_{(n)(i+1)}, rl_{(n-1)(i)} \}$.

One particle which belongs to a ring layer $rl_{(k)}$ exchange information with other layers $rl_{(k-1)}$ and $rl_{(k+1)}$ in order to make use of the social information achieved by the other layers. If one particle just exchanges information with the same neighbourhood through the entire execution of the algorithm, the neighbourhood would be static. In this case, the topology is just a simple *local* approach. To avoid this static possibility, one important process was added to the proposed topology, called *rotation skill.*

3.3 Ring layer rotation skill

One of the biggest problems of PSO is the stagnation. In nonlinear multimodal problems, which have many local minima regions, the swarm can be stagnated in one those regions of the search space during the algorithm execution. In those regions, the particles can spend many time steps without varying the value of the best position found by their neighbourhoods. It can cause a stagnation state in the entire swarm. Hence, the proposed strategy to avoid stagnation is to add the ability of rotation to each ring layer. By adding the rotation skill, the position of the particles in the ring can be altered. Hence, the neighbourhood is changed. It can move the swarm from one region to another, increasing the exploration of the space search by considering different local minima regions.

In the Multi-Ring approach, the particles of different rings exchange information between themselves using local communication. The particle that belongs to a ring $rl_{(k)}$ exchanges information with other rings $rl_{(k-1)}$ and $rl_{(k+1)}$ aiming to transmit its previous experience.

Each ring layer of the Multi-Ring has the ability to self-rotate. It means that if the ring layer $rl_{(k)}$ does not improve its own best position, defined as *ringbest*, it will be rotated. In the rotation process, each particle in the ring layer has its index changed to $i = (i+d) \bmod (nl)$, where d is the rotation distance and nl is the number of particles in a layer. Thus, the new neighbourhood will be composed by $\{ rl_{(k)(i-1)}, rl_{(k)(i+1)}, rl_{(k-1)(i+d)}, rl_{(k+1)(i+d)} \}$. One can note that the main difference is the communication replacement between the neighbour ring layers.

Figure 4 shows an example of the rotation process. Initially, the particle E communicates with {D,F,B,H}. After the rotation, the neighbourhood of the particle E is changed to {D,F,A,G}. This process improves the social information exchange. Besides, it also improve the convergence capacity of the swarm. One can note that all the particles of the same ring layer change its neighbourhood.

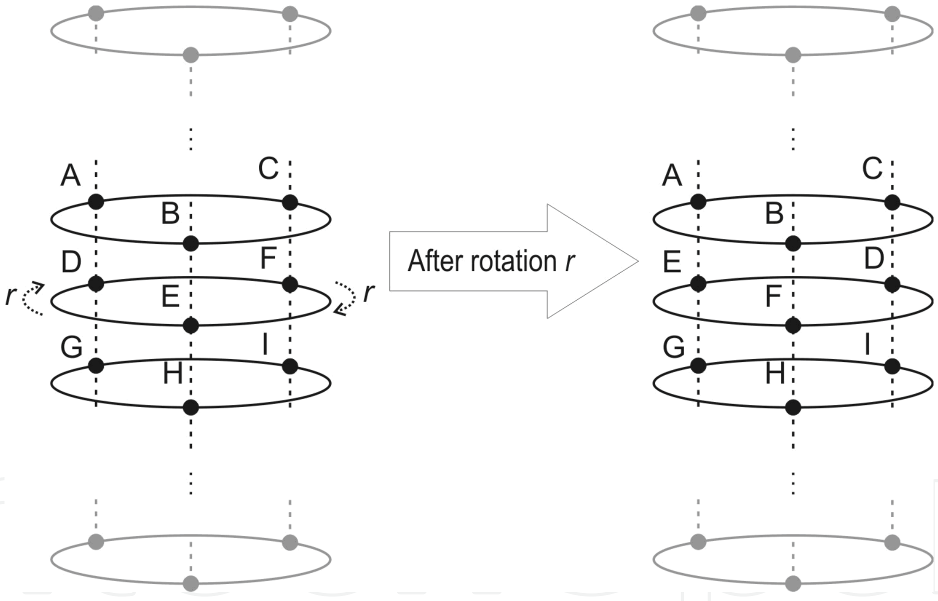


Fig. 4. Rotation skill example.

The main mechanism of the rotation process is the trigger criterion that will be used to invoke the rotation skill. It is based on the idle status of the best solution found in a specific ring layer, i.e. the number of iterations that the best solution found by a specific ring does not change. To clarify the process described above, consider a ring layer $rl_{(k)}$. If the best solution found remains with the same value during t_r iterations, so this ring layer will be rotated after t_r iteration.

Therefore, the use of the exploration capacity provided by the local communication combined with rotation mechanisms helps to avoid the stagnation of the solution found by the swarm during the execution of the PSO algorithm.

3.4 Multi-ring PSO algorithm

Inspired by the entity described in the previous section, the PSO algorithm with this novel topology has some new specific steps that must be considered in the execution process of the algorithm. Figure 5 illustrates the Multi-Ring topology containing an example with two ring layers ($rl_{(1)}$ and $rl_{(2)}$) of three particles.

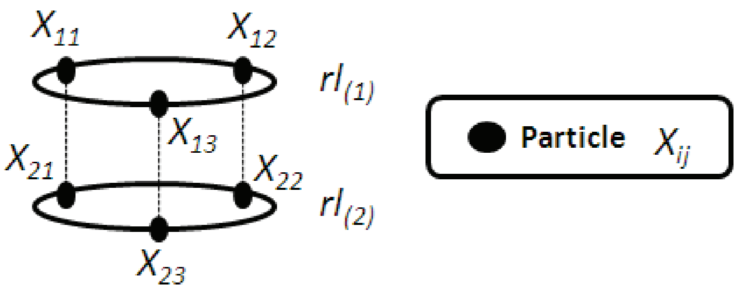
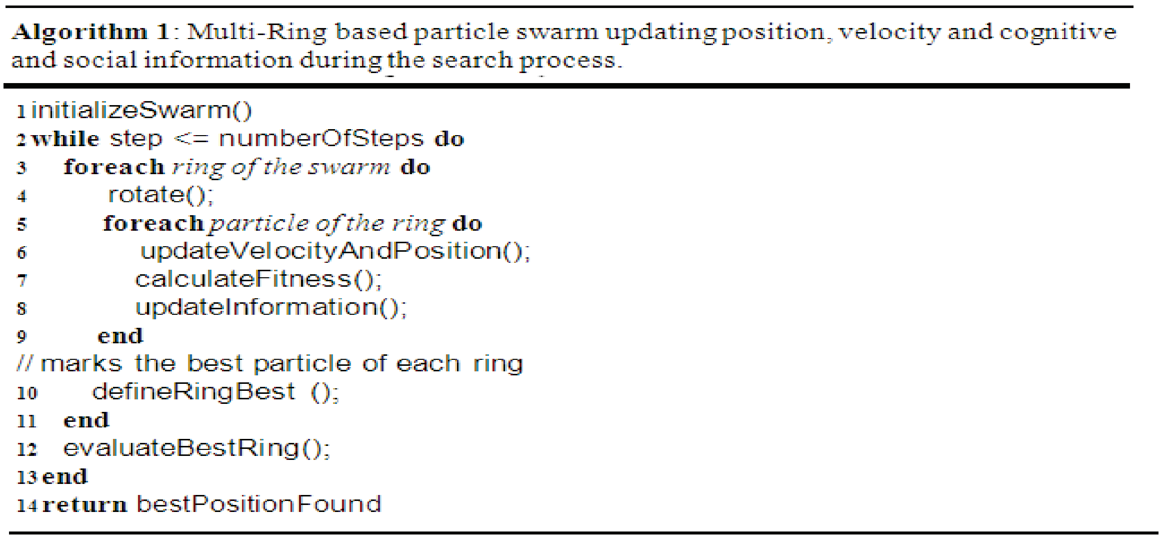


Fig. 5. Multi-Ring topology of 2 ring layers with 3 particles.

In the basic PSO algorithm, one of the steps is the definition of the initial *pbest* and *gbest,* which represent the best position found by the particle and swarm, respectively. The manner how the value of *pbest* is updated or chosen is not influenced by the use of Multi-Ring topology, since it is related to the individual experience of the particle. Therefore, all the particles of the swarm define the *pbest* in the same way.

For each iteration, each ring layer performs a search and marks the particle that reached the best position of the entire ring layer. Different from *gbest,* in the Multi-Ring approach, the definition of this new position called *ringbest,* considers only the particles within the respective ring layer. Therefore, each ring layer will have its own *ringbest.* After that, the *ringbests* of each ring layer are compared with each other, and the best one will be selected as the best solution. The pseudo-code of the algorithm is presented in Figure 6.



Algorithm 1: Multi-Ring based particle swarm updating position, velocity and cognitive and social information during the search process.

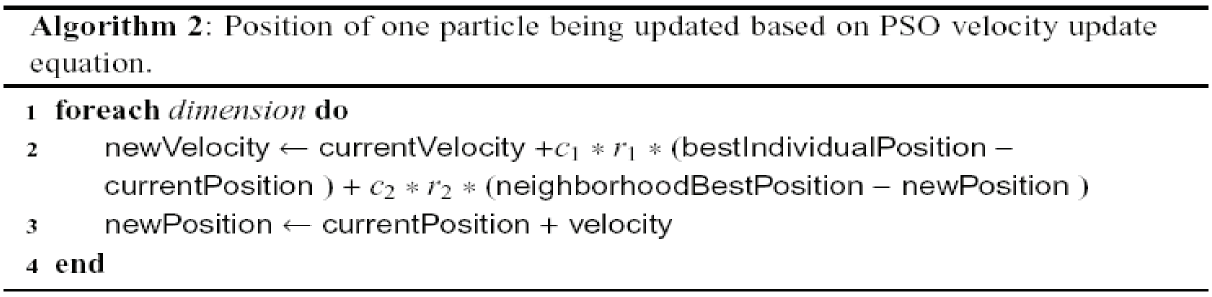
```
initializeSwarm()
while step <= numberOfSteps do
    foreach ring of the swarm do
        rotate();
        foreach particle of the ring do
            updateVelocityAndPosition();
            calculateFitness();
            updateInformation();
        end
// marks the best particle of each ring
        defineRingBest ();
    end
    evaluateBestRing();
end
return bestPositionFound
```

Fig. 6. Multi-Ring particle swarm optimization algorithm.

After the *pbest* evaluation, the algorithm performs the selection of the best particle of the respective ring layer. This process is repeated for all the ring layers that compose the Multi-

Ring topology (see Line 10 in Figure 6). Line 12 of Figure 6 shows that the selection of the best *ringbest* for the topology occurs at the end of each algorithm iteration.

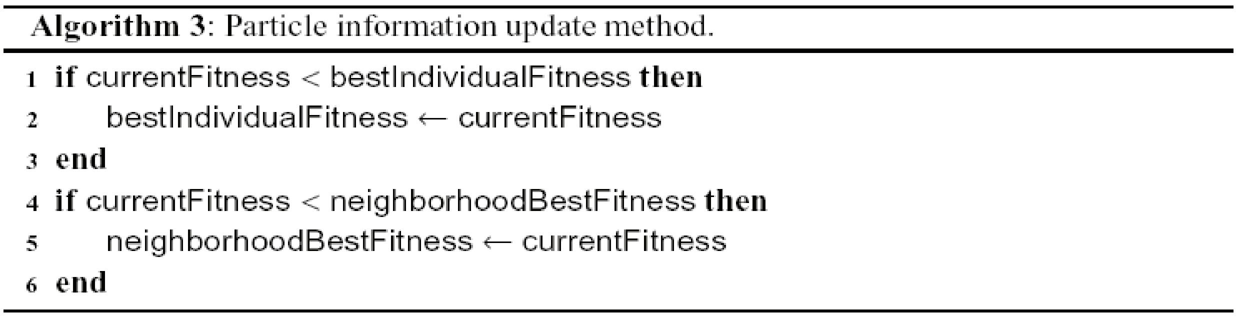
The **updateVelocityAndPosition()** and **updateInformation()** processes are the core of any PSO algorithm. These two processes are described in the pseudo-codes of Algorithms 2 and 3 that are shown in Figures 7 and 8, respectively.



Algorithm 2: Position of one particle being updated based on PSO velocity update equation.

```
foreach dimension do
    newVelocity ← currentVelocity +$c_1 * r_1 *$ (bestIndividualPosition –
    currentPosition ) + $c_2 * r_2 *$ (neighborhoodBestPosition – newPosition )
    newPosition ← currentPosition + velocity
end
```

Fig. 7. Particle swarm position and velocity updating algorithm.



Algorithm 3: Particle information update method.

```
if currentFitness < bestIndividualFitness then
    bestIndividualFitness ← currentFitness
end
if currentFitness < neighborhoodBestFitness then
    neighborhoodBestFitness ← currentFitness
end
```

Fig. 8. Particle information updating algorithm.

The other process involved in Algorithm 1 execution is **rotate()** (see Line 4), which belongs only to the multi-ring topology approach proposed here. One can note that this step is done before the *fitness* evaluation. Figure 9 can be used to represent this process. Consider that the rotation trigger criterion used is the number of successive iterations that the best solution found (*ringbest)* by the ring layer has not improved and *d* is the number of positions that the ring will be rotated. Using t_r to represent the trigger mechanism and X_i to represent the particle *i* of the ring layer X, the ring layer X will be rotated if the value of its best position has not improved after t_r iterations. Otherwise, if the respective number of iterations is below $t_{r,}$ the algorithm does not rotate the ring layer X and the number of iterations is incremented. This process is done for all ring layers of the Multi-ring topology.

When the trigger criterion is satisfied, the ring layer is rotated by *d* particles and the number of iterations that the *ringbest* did not improved is reset. It implies in a new neighbourhood for each particle of the rotated ring and, as a result, it improves the social information exchange. One can note that there is a trigger criteria $t_{r(k)}$ for each ring layer $rl_{(k)}$. The figure 10 illustrates the rotation algorithm process described above.

4. Dispersed particle swarm optimization

4.1 Dispersed PSO concepts

In particle swarm optimization (PSO) literature, the published social coefficients settings aims to increase the search around the swarm memory. However, few approaches

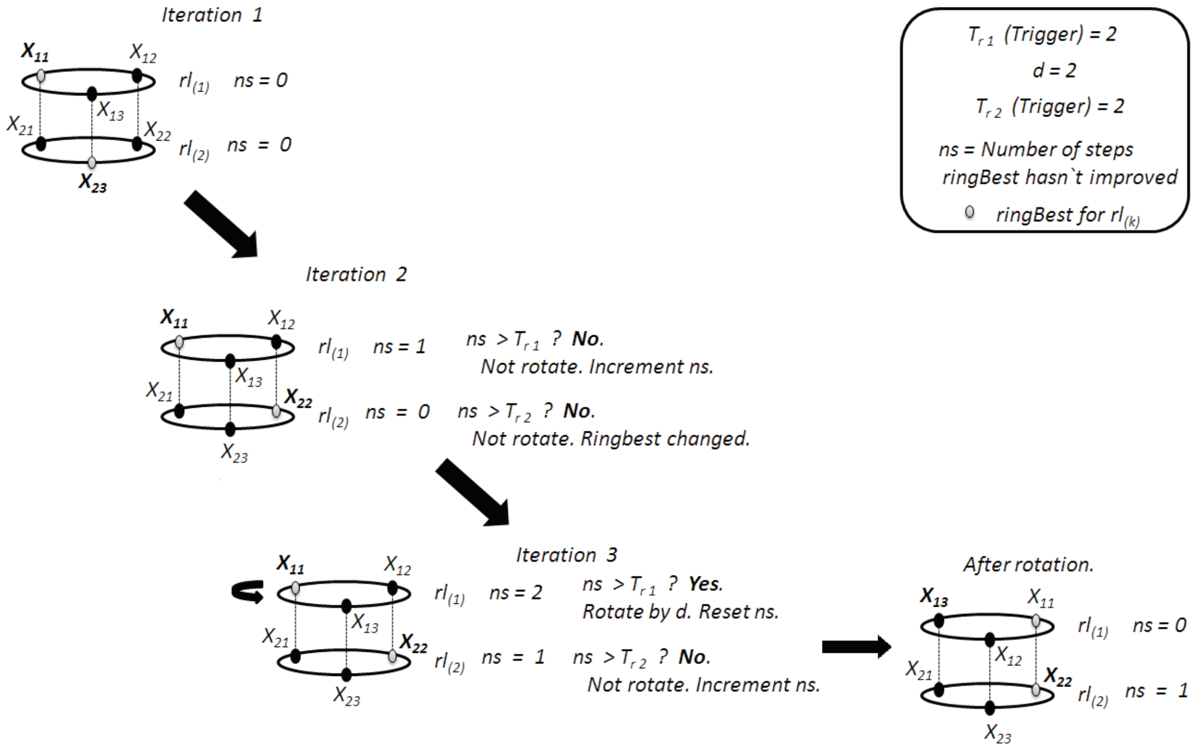
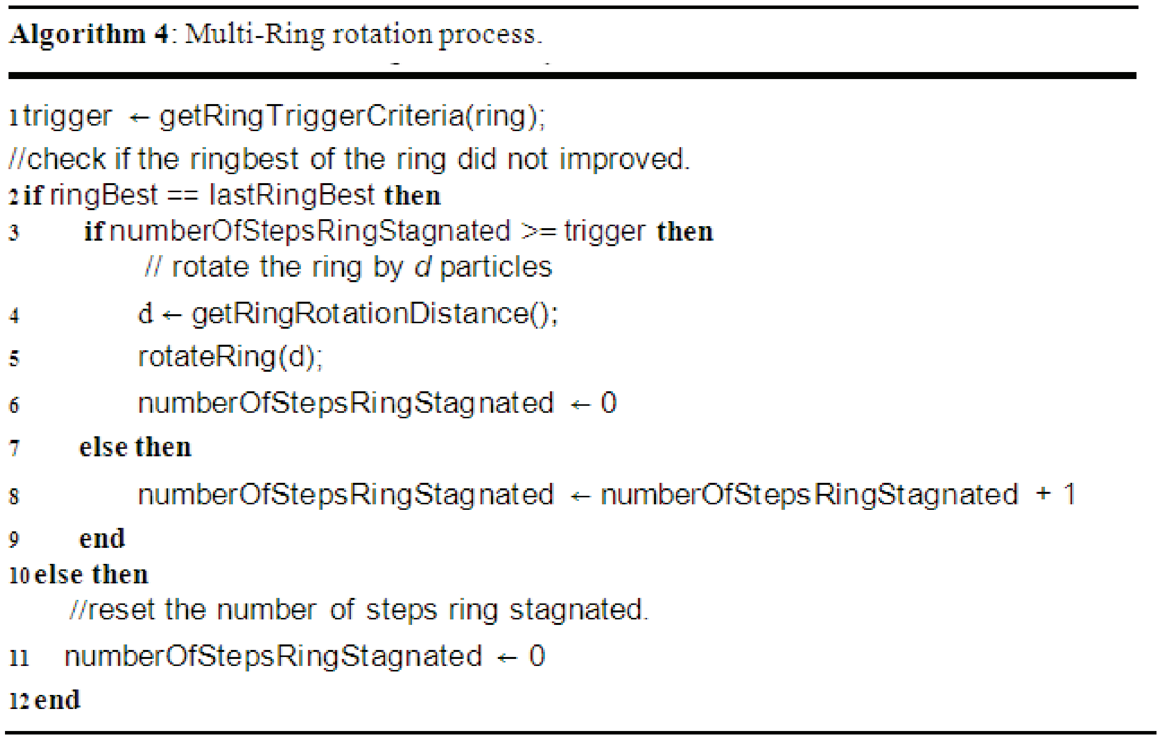


Fig. 9. Rotation simulation process.



Algorithm 4: Multi-Ring rotation process.

```
trigger ← getRingTriggerCriteria(ring);
//check if the ringbest of the ring did not improved.
if ringBest == lastRingBest then
    if numberOfStepsRingStagnated >= trigger then
        // rotate the ring by d particles
        d ← getRingRotationDistance();
        rotateRing(d);
        numberOfStepsRingStagnated ← 0
    else then
        numberOfStepsRingStagnated ← numberOfStepsRingStagnated + 1
    end
else then
    //reset the number of steps ring stagnated.
    numberOfStepsRingStagnated ← 0
end
```

Fig. 10. Multi-Ring rotation process.

take into account the useful information inside the particle's memories. Thus, to improve the convergence speed, a new approach considering the social coefficient was proposed (Cai et al, 2008) by introducing an explicit selection pressure, in which each particle decides its search direction toward the personal memory or swarm memory.

The fitness value of the current position of each particle represents its adaptive capability, and it can be selected to distinguish between search modes. In this approach, the better the fitness of the current position is, more adaptation this particle can get to enhance its survival ratio.

Since the literature consider the extreme situations such as *global* or *local* approaches, they neglect the differences between the particles in the search process. These settings lose some information that can be useful to find the global optima or escape from local optima.

To minimize this problem, Cai *et al*. (Cai *et al*., 2008) proposed a new index to indicate performance differences, named *Grade$_i$(t)*. The definition of the index is provided as follows:

$$Grade_i(t) = \frac{f_{worst}(t) - f\left[\vec{x}_i(t)\right]}{f_{worst}(t) - f_{best}(t)}, \tag{2}$$

where $f_{worst}(t)$ and $f_{best}(t)$ are the worst and the best fitness values of the swarm at time step *t*, respectively.

Occasionally, the swarm can converge onto one point, that means $f_{worst}(t) = f_{best}(t)$. In this case, the value *Grade$_i$(t)* of an arbitrary particle *i* is set to 1. *Grade$_i$(t)* represents the performance of particle *i* at time *t*, according to its fitness value of the current position. The better particle is, the larger *Grade$_i$(t)* is, and vice-versa.

In most cases, if the fitness value of particle *i* is better than the fitness of particle *u*, the probability of *i*'s neighbourhood falls into a global optima is higher than the particle *u*. Thus, the particle *i* should give more attentions to exploit its neighbourhood. On the contrary, it may tend to explore other regions with a larger probability. Therefore, the best solution should perform a complete search around its historical best position and the worst solution should perform a global search around the best solution found so far.

Then, the dispersed social coefficient of particle *i* at time *t* is set as follows:

$$c_{2,i}(t) = c_{low} + \left(c_{up} - c_{low}\right) Grade_i(t), \tag{3}$$

where c_{up} and c_{down} are two predefined numbers, and $c_{2,i}(t)$ represents the social coefficient of particle *j* at time *t*. One can note that better particles will present higher values for c_2.

4.2 Multi-ring dispersed particle swarm optimization

Based on the new manner to determine the acceleration coefficients, the dispersed PSO approach was included into the Multi-Ring Topology. By applying the *Grade$_i$(t)* to determine the communication rules inside each ring, we expect that it will improve the performance of the search made by the Multi-Ring PSO algorithm.

The grade criterion, which is calculated by evaluating a fitness comparison, will make the best solutions perform searches around its vicinities and the worst solutions look for the history of a larger part of the swarm, in order to increase its experience.

Figure 11 illustrates the grade strategy in action. If the *Grade$_i$(t)* is above a threshold (good solutions), we reduce the neighbourhood of the particle. In this case, the neighbourhood of the particle *i* is defined as { $rl_{(k)(i-1)}$, $rl_{(k)(i+1)}$, $rl_{(k-1)(i)}$, $rl_{(k+1)(i)}$ }. Otherwise, the particle should expand its communication range to achieve more information, which it would get information from all particles in the same ring in order to increase its experience. One can note that rings which the *ringbest* are below the threshold will perform searches using a more connected structure in order to achieve better regions in the search space. In spite of this, one should note that the communication between the ring layer remains local.

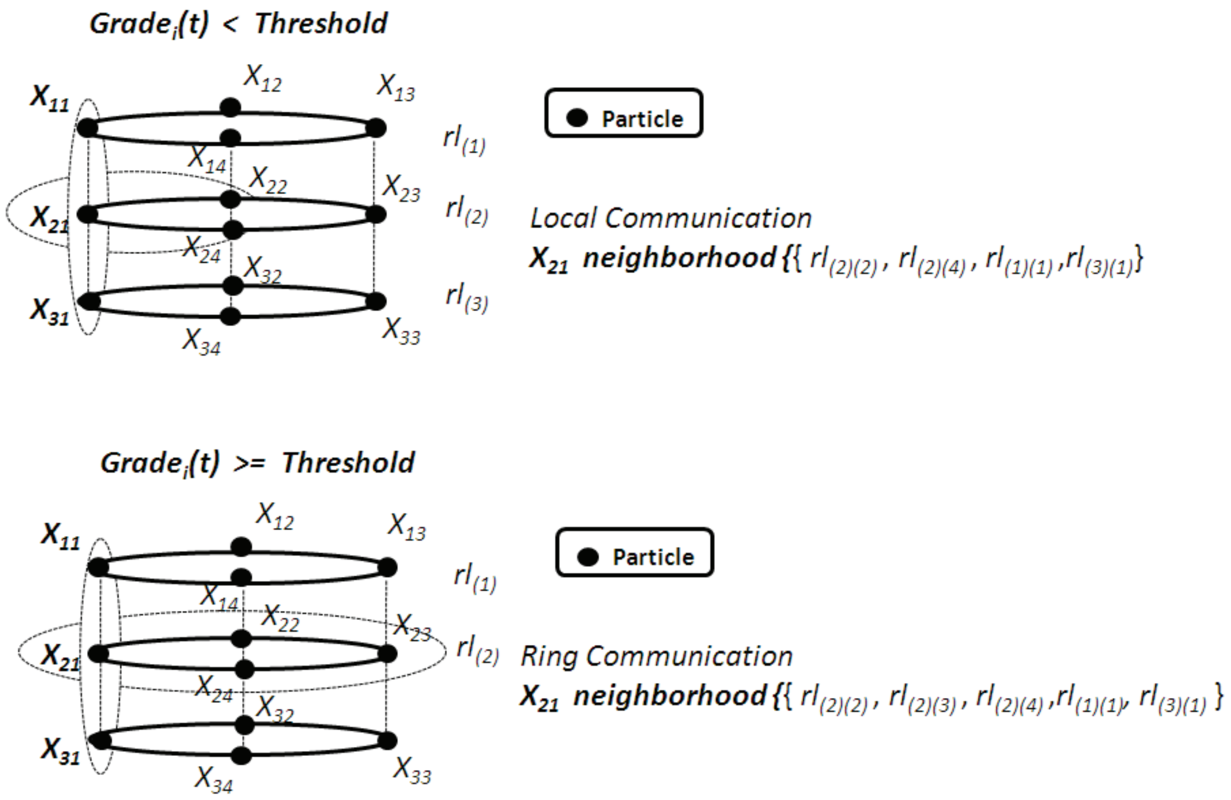


Fig. 11. Grade criteria at Multi-Ring PSO.

5. Simulation setup

In Table 1, seven commonly used benchmark functions are presented. Their names, definitions, search space, initialization range, and optimum compose the columns of Table 1. All the functions described are used as minimization problems classified as unimodal (f_1) or multimodal (f_2, f_3, f_4, f_5, f_6 and f_7) containing many local minima and a high level of complexity. For the simulations, each function was tested with 30 dimensions.

Each simulation was performed using PSO with inertia factor. The reduction of w is linear and starts at 0.9 decreasing to 0.4 along the iterations. The number of rings was varied based on a swarm with 30 particles. Thus, the number of ring layers is defined as $nl = 30 / n$. The rotation distance is equal to a half of the ring layer size ($d = nl / 2$) and the trigger criterion was $t_r = 15$.

We used c_1 and c_2 equal to 2.05. In the MRDPSO, we used $c_{low} = 1.5$ and $c_{up} = 2.5$.

The number of iterations used in a single simulation was 10,000. After 30 executions, the mean and standard deviation were stored. The initial positions of the entire swarm are defined randomly inside the initialization range of the benchmark function used. Comparisons with other topologies were also provided for convergence performance validation in both inertia and dispersed approaches.

Function	Equation	Search space	Init.	Opt.
Rosenbrock (f_1)	$f_1 = \sum_{i=1}^{n-1}\left[100\left(x_{i+1} - x_i^{\,2}\right)^2 + (1 - x_i)^2\right]$	$(-30, 30)^D$	$(15, 30)^D$	1.0^D
Ackley (f_2)	$f_4 = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^{\,2}}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$(-32, 32)^D$	$(16, 32)^D$	0.0^D
Griewank (f_3)	$f_3 = 1 + \sum_{i=1}^{n}\frac{x_i^{\,2}}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$(-600, 600)^D$	$(300, 600)^D$	0.0^D
Penalized P8 (f_4)	$f_7 = \frac{\pi}{n}\{10\sin^2(\pi y_i)$ $+\sum_{i=1}^{n-1}(y_i - 1)^2\{1 + 10\sin^2(\pi y_{i+1})\}$ $+(y_n - 1)^2\} + \sum_{i=1}^{n}\mu(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $\mu(x_i, a, k, m) = \left\{\begin{array}{ll} k(x_i - a)^m & , \quad x_i > a \\ 0 & , \quad -a \leq x_i \leq a \\ k(-x_i - a)^m & , \quad x_i < a \end{array}\right\}$	$(-50, 50)^D$	$(25, 50)^D$	-1.0^D
Penalized P16 (f_5)	$f_8 = 0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^{n-1}(x_i - 1)^2\{1 + \sin^2(3\pi x_{i+1}$ $+(x_n - 1)^2 \times \{1 + \sin^2(2\pi x_n)\}\} + \sum_{i=1}^{n}\mu(x_i, 5, 100, 4)$	$(-50, 50)^D$	$(25, 50)^D$	1.0^D
Rastrigin (f_6)	$f_2 = 10n + \sum_{i=1}^{n}\left[x_i^{\,2} - 10\cos(2\pi x_i)\right]$	$(-5.12, 5.12)^D$	$(2.56, 5.12)^D$	0.0^D
Schwefel 2.6 (f_7)	$f_6 = -\sum_{i=1}^{n} x_i \sin\left(\sqrt{x_i}\right)$	$(-500, 500)^D$	$(-500, -250)^D$	420.96^D

Table 1. Benchmark functions used in the experiments.

6. Results

This section shows the results of the experiments involving the benchmark functions described previously. In the first section (Section 6.1) of the simulation results, we vary the number of ring layers. The second one (Section 6.2) presents a comparison between each Multi-ring approach and the topologies already mentioned in Section 2 of this paper.

6.1 Ring layer variation

Table 2 shows the results of MRPSO varying the number of ring layers in simulations involving each benchmark function. For function f_6, the structure composed by 3 ring layers behaved better. The 5-ring MRPSO achieved a better performance for functions f_1, f_2 and f_3, and the 10-ring behaves better for f_4, f_5, and f_7.

One can note that the best choice for function f_6 was 3-ring layers followed by the 5-rings configuration. Thus, in general, the 5-rings configuration achieved the most satisfactory fitness values. This is due to the fact that maintaining a balance between the degree of

Function	MRPSO					
	3 ring layers		5 ring layers		10 ring layers	
	Mean	SD	Mean	SD	Mean	SD
(f_1)	17.48	9.39	**15.05**	**7.87**	29.48	29.33
(f_2)	3.19E-9	3.40E-9	**2.28E-9**	**3.53E-9**	2.56E-10	3.41E-10
(f_3)	0.0097	0.01	**0.00509**	**0.0063**	0.0062	0.00902
(f_4)	2.56E-32	3.1E-32	2.72E-32	4.26E-32	**1.62E-32**	**2.23E-32**
(f_5)	2.72	2.82	3.56	4.94	**0.56**	**1.03**
(f_6)	**16.3**	**5.19**	17.36	7.41	19.57	5.85
(f_7)	3305.57	577.24	2963.11	498.9	**2686.44**	**456.16**

Table 2. Number of ring layers variation of MRPSO.

communication inside a ring and through different rings, the MRPSO can reach regions with better solutions.

For the dispersed approach for the proposed topology (MRDPSO) (see Table 3) the 5-rings configuration achieved the best overall performance.

Function	MRDPSO					
	3 ring layers		5 ring layers		10 ring layers	
	Mean	SD	Mean	SD	Mean	SD
(f_1)	12.99	19.05	9.84	14.42	**8.44**	**5.97**
(f_2)	**6.83E-15**	**1.44 E-15**	7.31E-15	9.01E-16	7.54E-15	1.61E-15
(f_3)	**0.70E-4**	**0.01**	0.80E-4	0.80E-4	0.01	0.01
(f_4)	4.08E-32	5.40E-32	4.55E-32	5.36E-32	**8.37E-33**	**7.96E-33**
(f_5)	**0.28**	**0.42**	0.29	0.41	0.37	0.78
(f_6)	14.725	4.008	**13.929**	**3.102**	20.39	7.54
(f_7)	4063.20	543.16	**3969.54**	**563.79**	3975.34	356.32

Table 3. Number of ring layers variation of MRDPSO.

For functions f_2, f_3 and f_5 the 3-rings configuration achieved the best fitness values. The configuration with 10 ring layers achieved better performance in functions f_1 and f_4. Again, we can consider that a configuration with 5-rings is enough to reach satisfactory fitness values in complex and different search spaces.

The dispersed coefficient improves the convergence speed, thus is clearly expected that MRDPSO will converge fast than MRPSO. Therefore, the 5-rings configuration can be considered the best configuration choice in both MRPSO and MRDPSO.

Varying the number of rings, both MRPSO and MRDPSO can fit to different kind of problems. Hence, the number of rings can be varied according to the problem to be solved.

6.2 Topologies comparison

The results between the comparison of MRPSO and MRDPSO and other topologies are shown in Table 4 and Table 5, respectively. For all the functions, the fitness values presented by the Multi-ring approaches are the ones that achieved the best performance or similar performance to the best performance.

f	MRPSO		Global		Local		Von Neumann		Four Clusters	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
(f_1)	**15.05**	**7.87**	17.712	6.3686	18.215	12.995	39.54	29.36	88.15	44.51
(f_2)	2.28E-9	3.53E-9	**1.26E-14**	**3.04E-15**	9.61E-8	8.70E-8	1.81E-10	1.86E-10	0.16	0.51
(f_3)	0.00509	0.0063	0.01	0.02	**0.0031**	**6.4E-3**	0.008	0.01	0.04	0.03
(f_4)	1.62E-32	2.23E-32	5.02E-32	5.52E-32	**6.22E-33**	**7.82E-33**	9.94E-33	1.27E-32	9.42E-32	5.05E-32
(f_5)	**0.56**	**1.03**	4.46	8.31	1.33	1.47	1.00	1.64	52.67	39.25
(f_6)	16.3	5.19	**15.35**	**5.93**	21.94	5.33	19.58	4.89	19.58	7.37
(f_7)	2686.44	456.16	**2164.17**	**368.74**	2964.79	389.31	2898.37	347.01	3633.34	519.53

Table 4. Comparison between MRPSO and other topologies.

f	MRDPSO		Global Dispersed		Local Dispersed		Von Neumann Dispersed		Four Clusters Dispersed	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
(f_1)	**8.44**	**5.97**	32.8099	33.7074	15.3180	18.0088	10.8321	13.8913	83.8264	54.0686
(f_2)	**6.83E-15**	**1.44 E-15**	1.13E-14	3.60E-15	9.73E-13	5.43E-13	7.54E-15	9.01E-16	0.2947	0.6127
(f_3)	0.70E-4	0.01	0.0134	0.0137	**4.09E-11**	**2.24E-10**	0.0103	0.014	0.0529	0.0498
(f_4)	8.37E-33	7.96E-33	4.86E-32	4.81E-32	**5.23E-33**	**7.53E-33**	2.72E-32	4.1436	9.10E-32	5.09E-32
(f_5)	0.28	0.42	0.7080	1.5170	**0.1375**	**0.22**	0.16	0.401	63.50	40.47
(f_6)	**13.929**	**3.102**	13.99	3.66	20.96	4.6	21.16	6.84	36.92	6.86
(f_7)	3969.54	563.79	**2320.08**	**355.31**	3610.09	461.8253	3889.64	503.86	3750.91	334.53

Table 5. Comparison between MRDPSO and other topologies.

For function f_1 and f_5 MRPSO reaches the best fitness values (F = 15.05 and F = 0.56). For all the other functions used, it reaches values close to the best performance achieved by some well known topologies. A good example is the performance for functions f_6 and f_7, in which the only topology that achieved fitness values better than the ones achieved by MRPSO was the global topology. Hence, considering that MRPSO is a local communication topology, the results presented by Table 4 are strongly satisfactory.

Table 5 shows the results of the MRDPSO and topologies enabled with the dispersion grade strategy. For function f_1, $f_{2,}$ and f_6 the MRDPSO achieves the best fitness values (F=8.44, F=6.83E-15 and F=13.929). For the functions f_3 and f_5, it reaches a good value for the mean fitness with low standard deviation, close to the best performance achieved by well known topologies. For the function $f_{7,}$ MRDPSO achieves a value for the mean fitness in the same order of magnitude of Dispersed PSO with local, Von Neumann and Four Clusters topologies. Therefore, considering all results shown at the Table 5, the MRDPSO achieves good solutions for both unimodal and multimodal problems.

7. Conclusions

This paper has proposed a novel topology for Particle Swarm Optimization based on a multi layer structure based on local communication (***lbest***). The Multi-ring PSO (MRPSO) and also its combination with the Dispersed PSO approach (MRDPSO) are thoroughly described. The first one introduces a new feature called *rotation skill* that minimizes the stagnation probability of the PSO algorithm. In the second one, some changes in the communication rules inside the rings were made by using the dispersion grade attribute of Dispersed PSO approach.

The Multi-ring approaches achieved better performance or close to well-known topologies used for comparison. This fact exalts how satisfactory the results involving MRPSO and MRDPSO were. By varying the number of rings, the size of each ring and how the rotation skill work, the proposed approach could be used to reach good quality solutions for both unimodal and multimodal problems. Besides, it helps to avoid the swarm of being trapped at local minima regions, making it moves to new regions of space search in order to help the PSO algorithm to converge to the best solution.

Due to its peculiar features, the proposed PSO topology is an interesting approach able to be applied in different kind of optimization problems.

8. References

Abido, M.A. (2001). Optimal design of power system stabilizers using particle swarm optimization. *IEEE Trans. Energy Conversion,* Vol. 17, No. 3, (September 2002) Pages: 7 (406–413), ISSN 0885-8969

Agrafiotis, D.K. & Cedeno, W (2002). Feature selection for structure-activity correlation using binary particle swarms. *J. Medicinal Chem.,* Vol. 45, No. 5, (February 2002) Pages: 9 (1098–1107)

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*, Springer-Verlag, ISBN 0-19-509971-0, Berlin, Germany

Bastos-Filho, C. J. A.; Caraciolo, M. P. ; Miranda, P. B. C. ; Carvalho, D. F. . Multi Ring Particle Swarm Optimization. *In: SBRN'2008 (the 10th Brazilian Symposium on Neural Networks)*, pp. 111-116, 2008, Salvador/BA. *Proceedings of* SBRN'2008 (the 10th Brazilian Symposium on Neural Networks), 2008

Beyer, H.-G. (2001). *The Theory of Evolution Strategies,* Springer-Verlag New York, Inc., ISBN 3-540-67297-4, New York, NY, USA

Beyer, H.-G. (2002). Evolution strategies: A comprehensive introduction. *Nat. Comput.,* Vol. 1, No. 1, (March 2002) Pages: 49 (3-52), ISSN 1567-7818

Bratton, D. & Kennedy, J. (2007). Defining a standard for particle swarm optimization, *Proceedings of Swarm Intelligence Symposium SIS 2007*, pp. 120–127, ISBN 1-4244-0708-7, Honolulu, HI, April 2007, IEEE Service

Cai, X., Cui, Z., Zeng, J. & Tan, Y. (2008). Dispersed particle swarm optimization. *Information Processing Letters,* Vol. 105, No. 6, (February 2008) Pages: 4 (231-235), ISSN 0020-0190

Carvalho, D.F. & Bastos-Filho, C.J.A. (2008). Clan particle swarm optimization, *Proceedings of IEEE Congress on Evolutionary Computation CEC 2008 (IEEE World Congress on Computational Intelligence)*, pp. 3044-3051, ISBN 978-1-4244-1822-0, Hong Kong, June 2008

Carvalho, D.F. & Bastos-Filho, C.J.A. (2009). Clan particle swarm optimization. *International Journal of Intelligent Computing and Cybernetics,* Vol. 2, No. 2, (June 2009) Pages: 31 (197 - 227), ISSN 1756-378X

Clerc, M. & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation,* Vol. 6, No. 1, (February 2002) Pages: 5 (58-73), ISSN 1089-778X

Coello Coello, C.A. & Lechuga, M.S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization, *Proceedings of 2002 IEEE Congr. Evolutionary Computation CEC 2002*, pp. 1051–1056, ISBN 0-7803-7282-4, Honolulu, HI, May 2002, IEEE Computer Society, Washington, DC, USA

Eberhart, R.C. & Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms, *Proceedings of 2001 IEEE Congr. Evolutionary Computation CEC 2001*, pp. 94–100, ISBN 0-7803-6657-3, Seoul, South Korea, May 2001

Engelbrecht, A.P. (2005). *Fundamentals of Computational Swarm Intelligence,* John Wiley & Sons, ISBN 0470091916

Fogel, D. (1996). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence,* Wiley-IEEE Press, ISBN 978-0-471-66951-7

Jiao, B., Lian, Z. & Gu, X. (2008). A dynamic inertia weight particle swarm optimization algorithm. *Chaos, Solitons and Fractals,* Vol. 37, No. 3, (August 2008) Pages: 7 (698–705), ISSN 0960-0779

Kennedy, J. & Eberhart, R.C. (1995). Particle swarm optimization, *Proceedings of the IEEE Int. Conf. on Neural Networks*, pp. 1942-1948, ISBN 0-7803-2768-3, Perth, WA, Australia, December 1995, IEEE Service Center, Piscataway, NJ

Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, *Proceedings of 1999 Congress on Evolutionary Computation CEC 99*, pp. -1938, ISBN 0-7803-5536-9, Washington, DC, USA, July 1999, IEEE Service

Kennedy, J. & Mendes, R. (2002). Population structure and particle swarm performance, *Proceedings of the 2002 Congress on Evolutionary Computation CEC 2002*, pp. 1671-1676, ISBN 0-7803-7282-4, Honolulu, HI, USA, May 2002, IEEE Press

Kennedy, J. (2005). Why does it need velocity?, *Proceedings of the Swarm Intelligence Symposium SIS 2005*, pp. 38-44, ISBN 0-7803-8916-6, Pasadena, California, June 2005

Mendes, R., Kennedy, J. & Neves, J. (2003). Watch thy neighbor or how the swarmcan learn from its environment, *Proceedings Swarm Intelligence Symposium SIS 2003*, pp. 88-94, ISBN 0-7803-7914-4, Indianapolis, Indiana, USA, April 2003, IEEE Service

Parsopoulos, K.E. & Vrahatis, M.N. (2001). Particle swarm optimizer in noisy and continuously changing environments, *Proceedings of Artificial Intelligence and Soft Computing*, pp. 289–294, Cancun, Mexico, 2001, IASTED/ACTA Press

Parsopoulos, K.E., Laskari, E.C. & Vrahatis, M.N. (2002). Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, Vol. 1, No. 2-3, (June 2002) Pages: (235–306), ISSN 1572-9796

Peer, E.S., van den Bergh, F. & Engelbrecht, A.P. (2003). Using neighbourhoods with the guaranteed convergence PSO, *Proceedings of 2003 IEEE Swarm Intelligence Symposium SIS '03*, pp. 235-42, ISBN 0-7803-7914-4, Indianapolis, Indiana, USA, April 2003, IEEE Service

Potter, M.A. & De Jong, K.A. (1994). A cooperative coevolutionary approach to function optimization, In: *The Third Parallel Problem Solving From Nature*, Pages: 8 (249–257), Springer Berlin / Heidelberg, ISBN 978-3-540-58484-1, Berlin, Germany

Praveen, K.; Sanjoy, D. & Stephen, M.W. (2007). Multi-objective hybrid PSO using µ-fuzzy dominance, *Proceedings of 9th annual conference on Genetic and evolutionary computation GECCO '07*, pp. 853–860, ISBN 978-1-59593-697-4, London, England, July 2007, ACM, New York, NY, USA

Santana-Quintero, L.V.; Coello Coello, C.A.; Hernandez-Diaz, A.G. & Velazquez, J.M.O. (2008). Surrogate-based Multi-Objective Particle Swarm Optimization, *Proceedings of Swarm Intelligence Symposium SIS 2008*, pp. 1-8, ISBN 978-1-4244-2704-8, St. Louis, Missouri, September 2008, IEEE

Schwefel, H.-P. (1981). Numerical Optimization of Computer Models, John Wiley & Sons, Inc, ISBN 0471099880, New York, NY, USA

Shi, Y. & Eberhart, R.C. (1998). A modified particle swarm optimizer, *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69-73, ISBN 0-7803-4869-9, Anchorage, AK, USA, May 1998, IEEE Press, Piscataway, NJ

Shi, Y. & Krohling, R.A. (2002). Co-evolutionary particle swarm optimization to solve min-max problems, *Proceedings of 2002 IEEE Conf. Evolutionary Computation CEC 2002*, pp. 1682–1687, ISBN 0-7803-7282-4, Honolulu, HI, May 2002, IEEE Computer Society, Washington, DC, USA

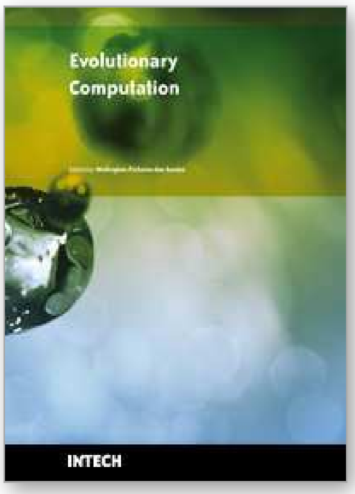
Suganthan, P.N. (1999). Particle swarm optimiser with neighbourhood operator, *Proceedings of the 1999 Congress on Evolutionary Computation CEC 99*, pp. -1962, ISBN 0-7803-5536-9, Washington, DC, USA, June 1999

Sutton, A.M.; Whitley, D., Lunacek, M. & Howe, A. (2006). PSO and multi-funnel landscapes: how cooperation might limit exploration, *Proceedings of 8th Annual Conference on Genetic and Evolutionary Computation GECCO '08*, pp. 75–82, ISBN 1-59593-186-4, Seattle, Washington, USA, July 2008, ACM, New York, NY, USA

van den Bergh, F. & Engelbrecht, A.P. (2001). Training product unit networks using cooperative particle swarm optimisers, *Proceedings of International Joint Conference on*

Neural Networks IJCNN '01, pp. 126–131, ISBN 0-7803-7044-9, Washington, DC, USA, July 2001

van den Bergh, F. & Engelbrecht, A.P. (2004). A Cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation,* Vol. 8, No. 3, (June 2004) Pages: 14 (225–239), ISSN 1089-778X



Evolutionary Computation
Edited by Wellington Pinheiro dos Santos

ISBN 978-953-307-008-7
Hard cover, 572 pages
Publisher InTech
Published online 01, October, 2009
Published in print edition October, 2009

This book presents several recent advances on Evolutionary Computation, specially evolution-based optimization methods and hybrid algorithms for several applications, from optimization and learning to pattern recognition and bioinformatics. This book also presents new algorithms based on several analogies and metafores, where one of them is based on philosophy, specifically on the philosophy of praxis and dialectics. In this book it is also presented interesting applications on bioinformatics, specially the use of particle swarms to discover gene expression patterns in DNA microarrays. Therefore, this book features representative work on the field of evolutionary computation and applied sciences. The intended audience is graduate, undergraduate, researchers, and anyone who wishes to become familiar with the latest research work on this field.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821