

Genetic Learning Particle Swarm Optimization with Interlaced Ring Topology

Bożena Borowska

Institute of Information Technology, Lodz University of Technology,
Wółczańska 215, 90-924 Lodz, Poland
bozena.borowska@p.lodz.pl

Abstract. Genetic learning particle swarm optimization (GL-PSO) is a hybrid optimization method based on particle swarm optimization (PSO) and genetic algorithm (GA). The GL-PSO method improves the performance of PSO by constructing superior exemplars from which individuals of the population learn to move in the search space. However, in case of complex optimization problems, GL-PSO exhibits problems to maintain appropriate diversity, which leads to weakening an exploration and premature convergence. This makes the results of this method not satisfactory. In order to enhance the diversity and adaptability of GL-PSO, and as an effect of its performance, in this paper, a new modified genetic learning method with interlaced ring topology and flexible local search operator has been proposed. To assess the impact of the introduced modifications on performance of the proposed method, an interlaced ring topology has been integrated with GL-PSO only (referred to as GL-PSOI) as well as with a flexible local search operator (referred to as GL-PSOIF). The new strategy was tested on a set of benchmark problems and a CEC2014 test suite. The results were compared with five different variants of PSO, including GL-PSO, GGL-PSOD, PSO, CLPSO and HCLPSO to demonstrate the efficiency of the proposed approach.

Keywords: genetic learning particle swarm optimization, enhanced diversity, particle swarm optimization, optimization;

1 Introduction

Developed by Kennedy and Eberhart [1, 2] particle swarm optimization (PSO) is a stochastic optimization method modeled on social behavior and intelligence of animal such as flocks of birds and fish schooling. Similar to other evolutionary methods, it is based on the population. The mechanism of the PSO method relies on particles following their best personal particle and globally the best particle in the swarm towards the most promising areas of the search space. Because of its easy implementation and high convergence rate, it is widely used in solving various optimization problems, including energetic [3], mechanics [4], scheduling problem [5], antenna design [6, 7], control systems [8], image classification [9] and many others. However likewise other evolutionary algorithms, PSO encounters some troubles including stagnation in local

adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

ICCS Camera Ready Version 2020

To cite this paper please use the final published version:

DOI: [10.1007/978-3-030-50426-7_11](https://doi.org/10.1007/978-3-030-50426-7_11)

optima, excessive loss of diversity and premature convergence [10]. A variety of different variants of PSO have been introduced to counteract these disadvantages and enhance the efficiency of PSO. Among them, the following improvements can be distinguished:

- Adjustment of basic coefficients. According to Shi and Eberhart [11], a key to the improvement of the PSO performance is inertia weight, which should be linearly decreased from 0.9 to 0.4. Clerc [12] recommended to use fixed factors, and indicates that inertia weight of 0.729 with fixed acceleration coefficients of 1.494 can enhance convergence speed. Five years later Trelea [13] proved that PSO with inertia weight of 0.6 and constant acceleration coefficients of 1.7 allowed to get faster convergence than that achieved by Eberhart [11] and Clerc [12]. The PSO method with nonlinear factors were proposed by Borowska [14,15]. Furthermore, the efficiency of changing factors was examined by Ratnawera et al. [16]. The cited authors concluded that time-varying acceleration coefficients (TVAC) helped to control local and global searching process more efficiently.
- Modification of the update equations. To improve searching process the researches propose to use a new update equation [17,18] or add a new component to existing velocity equation [19]. Another approach is to introduce, for ineffective particles, a repair procedure [10] with other velocity updating equations that helps more precisely determine swarm motion and stimulate particles when their efficiency decreases.
- Topology structure. According to Kennedy [20] topology structure affects the way information exchange and the swarm diversity. Many different topological structures have been proposed including: square, four clusters, ring, pyramid and the von Neumann topology [20-23]. Another approach is a multi-swarm structure recommended by Liang and Suganthan [24] and Chen et al. [25]. In contrast, Gong et al. [22] have introduced a two-cascading-layer structure. In turn, Wang et al. [26] developed PSO based on multiple layers.
- Learning strategy. It is used to improve performance of algorithm by breeding high quality exemplars from which other swarm particles can acquire knowledge and learn to search space. A multi-swarm PSO based on dynamic learning strategy has been presented by Ye et al.[27]. Likewise, Liang et al.[28] has proposed a comprehensive learning strategy (CLPSO) according to which, particle velocity is updated based on historical best information of all other particles. To greater improve the performance and adaptability of CLPSO, Lin et al. [29] recommend to use an adaptive comprehensive learning strategy with dynamically adjusting learning probability level according to the performance of the particles during the optimization process. Another approach is based on social learning PSO as described by Cheng et al. [30].
- Hybrid methods combine beneficial features of two or more approaches. They are used to strength PSO efficiency and achieve faster convergence as well as better accuracy of the resultant solution. Holden et al. [31] have proposed to join PSO with an ant colony optimization method. Li et al.[32] have combined PSO with jumping mechanism of SA (simulated annealing). A modified version based on

PSO and SA has been developed by Shieh et al. [33]. In turn, PSO with chaos has been presented by Tian and Shi[34], whereas Chen et al. [35] have proposed learning PSO based on biogeography. Furthermore, a hybrid approach based on improved PSO, cuckoo search and clustering method has been developed by Bouer and Hatamlou [36].

In order to enhance the PSO performance, Gong et al. [22] have merged the latter two categories and proposed genetic learning particle swarm optimization (GL-PSO). In GL-PSO, except PSO and genetic operators, a two layer structure have been applied in which the former is used to generate exemplars whereas the latter to update particles through the PSO algorithm.

The GL-PSO method improves the performance of PSO by constructing superior exemplars from which individuals of the population learn to move in the search space. Unfortunately, this approach is not free from disadvantages. In fact, the algorithm can achieve high convergence rate but in case of complex problems, due to global topology, the particle diversity quickly decreases and, as a result, impairs the exploration capability.

In order to enhance the diversity and adaptability of GL-PSO as well as to improve its performance in solving complex optimization problems, in this paper, a new modified genetic learning method, referred to as GL-PSOIF, has been demonstrated. The proposed GL-PSOIF method is based on GL-PSO in which two modifications have been introduced. Specifically, instead of global topology, an interlaced ring topology has been introduced. The second modification relies on introducing a flexible local search operator. The task of the interlaced ring topology is to increase the population diversity and improve effectiveness of the method by generating better quality exemplars. In turn, a flexible local search operator has been introduced to enrich searching and improve the exploration and the exploitation ability. To evaluate the impact of the proposed modifications on performance of the proposed method, the interlaced ring topology has been first integrated with GL-PSO only (referred to as GL-PSOI) and then together with a flexible local search operator (referred to as GL-PSOIF). Both methods were tested on a set of benchmark problems and a CEC2014 test suite [38]. The results were compared with five different variants of PSO, including the genetic learning particle swarm optimization (GL-PSO) [22], the comprehensive particle swarm optimizer (CLPSO) [28], the standard particle swarm optimization (PSO), the global genetic learning particle swarm optimization (GGL-PSOD) [23], and the heterogeneous comprehensive learning particle swarm optimization (HCLPSO) [39].

2 The PSO method

The PSO method was inspired by the social behavior of flocks of organisms (bird flocking, fish schooling, bees swarm) living in their natural environment [2, 3]. Likewise other evolutionary method, PSO is based on a population. Individuals of the population are called particles, and the population itself is called a swarm. In the PSO, the optimisation process is achieved by migration the particles towards the most

promising area of the search space. Assuming that migration occurs in the D -dimensional search space, we can imagine particle swarm as a set of points each of which possess knowledge about: its actual position described by the position vector $x_j=(x_{j1},x_{j2},\dots,x_{jD})$, its current speed of movement described by velocity vector $v_j=(v_{j1},v_{j2},\dots,v_{jD})$, its best position encountered by itself described by $pbest_j=(pbest_{j1},pbest_{j2},\dots,pbest_{jD})$, and the best position encountered in all swarm described as $gbest=(gbest_1,gbest_2,\dots,gbest_D)$. In the first iteration, the position vector value and the velocity vector value are randomly generated. In subsequent iterations, values of the vectors are updated based on the knowledge and acquired experience of the particles. Changing of the particles velocity is achieved based on the equation (1).

$$v_j(l+1) = w \cdot v_j(l) + c_1 \cdot r_1(pbest_j - x_j(l)) + c_2 \cdot r_2(gbest - x_j(l)) \quad (1)$$

Changing the particle position is realized by adding its actual velocity to its previous position (2)

$$x_j(l+1) = x_j(l) + v_j(l+1) \quad (2)$$

where: w - inertia weight, $pbest_j$ -the best j particle position, $gbest$ - the best position in a swarm, r_1, r_2 -random numbers generated from (0,1), c_1, c_2 - acceleration coefficients.

3 Genetic learning particle swarm optimization

In contrast to PSO, the GL-PSO algorithm possess a two-cascading-layer structure. One layer is used to generate exemplars, the other to update particles position and velocity through the PSO algorithm. To generate exemplars, three operators (crossover, mutation and selection) of the GA algorithm [37] are applied.

Exemplars e_j are selected from offspring. To generate offspring o_j for each dimension of particle j , a crossover operator is applied according to the formula:

$$o_j = \begin{cases} r \cdot pbest_j + (1-r) \cdot gbest, & \text{if } f(pbest_j) < f(pbest_k) \\ pbest_k & \text{otherwise} \end{cases} \quad (3)$$

where k is random selected particle, r -random number from (0,1).

Next, for each dimension, a random number $r \in [0,1]$ is generated and then if $r < p_m$, (where p_m probability mutation) the offspring is mutated. Then the offspring undergoes the selection operation according to the formula:

$$e_j \leftarrow \begin{cases} o_j, & \text{if } f(o_j) < f(e_j) \\ e_j, & \text{otherwise} \end{cases} \quad (4)$$

The particle velocity is updated based on the following equation:

$$v_j(l+1) = w \cdot v_j(l) + c \cdot r (e_j - x_j(l)) \quad (5)$$

where e_j is the exemplar of the j particle.

4 The proposed method

In order to improve the performance of global genetic learning particle swarm optimization (GL-PSO), in this article two modifications have been proposed: interlaced ring topology and flexible local search operator.

4.1 Interlaced ring topology

One of the main reason for inability to obtain and pursue satisfactory performance of the GL-PSO is the lack or weaknes ability to maintain diversity of the population (swarm). This leads to a loss of balance between exploration and exploitation and consequently to premature convergence and unsatisfactory results. To avoid this, it is necessary to develop tools that could help increase adaptability of the algorithm, which, in turn, should give satisfactory results.

Lin et al. [22] have introduced ring and a global learning component with linearly adjusted control parameters to enhance a GL-PSO diversity. This improves the adaptability of the method but is not sufficient. Hence, the problem remains open and other solutions should be sought. To improve the adaptability of the GL-PSO, in this paper, instead of global learning, the interlaced ring topology has been proposed. This approach uses two neighbour particles, like in the ring topology, but in every next iteration (except the first one), the order of the particles is changed as follows. The particle collection is divided into two parts (sets) and particles of the second part take up spaces between the particles of the first part alternately (one particle from the first set, another particle from the second set, and next one from the first set, another from the second set etc....) according to equation 6 and 7.

$$n_j = \frac{j+1}{2} \quad \text{for odd } j \quad (6)$$

$$n_j = \frac{j+N}{2} \quad \text{for even } j \quad (7)$$

where n_j is the position of the particle to be moved to the j place in the ring, $j=1 \dots N$, N is a swarm size (for example $n_2 = 5$ means that the second position in the ring is occupied by a particle from 5th place in the swarm).

Then, the position of exemplars are generated according to equations 8-10.

$$o_j = r \cdot pbest_{n_{j1}} + (1 - r) \cdot pbest_{n_{j2}} \quad (8)$$

$$n_{j1} = \begin{cases} N, & j = 1 \\ j - 1, & j > 1 \end{cases} \quad (9)$$

$$n_{j2} = \begin{cases} 1, & j = N \\ j + 1, & j < N \end{cases} \quad (10)$$

where according to the ring topology n_{j1} and n_{j2} are the indexes of the adjacent particles from the left and right side of the particle j , respectively.

4.2 Flexible local search operator

To improve the searching behavior of PSO and improve the exploitation capacity of the swarm, a flexible local search operator is introduced. The particle positions are updated according to the formula:

$$x_j^{k+1} = \begin{cases} pbest_j \cdot (1 + N(0,1)), & \text{otherwise} \\ x_j^k + v_j^{k+1}, & \text{if } p < s \end{cases} \quad (11)$$

where p is a randomly selected number in the range $[0,1]$, s is a real number linearly increasing from 0.6 to 0.8. This means that each particle has a 40 to 20% possibility to perform search in the vicinity of its personal best position. This means that, according to [16], in the early stage of the optimization process, the exploration is enhanced, and the local exploitation in the latter stage is facilitated.

5 Test results

In order to investigate the efficiency of the proposed modifications, the GL-PSOI (in which only the interlaced ring topology was adopted) and GL-PSOIF (with interlaced ring topology and flexible local search operator) were evaluated, separately. Both strategies were tested on a set of classical benchmark problems, and on the CEC2014 test suite. Twelve of them (6 selected benchmark function and 6 CEC2014 functions) are described in this article and depicted in Tables 1 and 2.

Table 1. Optimization test functions.

Function	Formula	f_{min}	Range	Accept
Sphere	$f_1 = \sum_{i=1}^n x_i^2$	0	$[-100, 100]^n$	10^{-5}
Schwefel	$f_2 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	0	$[-100, 100]^n$	100
Rosenbrock	$f_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	0	$[-5, 5]^n$	10^{-5}
Rastrigin	$f_4 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0	$[-32, 32]^n$	10^{-5}
Ackley	$f_5 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	0	$[-600, 600]^n$	10^{-5}
Penalized	$f_6 = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})]\}$	0	$[-50, 50]^n$	10^{-5}

$$+(x_n - 1)^2 [1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4),$$

$$u(z, a, k, m) = \begin{cases} k(z - a)^m, & z > a, \\ 0, & -a \leq z \leq a, \\ k(-z - a)^m, & z < -a. \end{cases}$$

Table 2. Selected CEC2014 test suite.

	Functions Name	Range	$F(x^*)$
F7	Rotated Bent Cigar Function	$[-100, 100]^n$	100
F8	Shifted and Rotated Rosenbrock's Function	$[-100, 100]^n$	400
F9	Shifted and Rotated Ackley's Function	$[-100, 100]^n$	500
F10	Shifted Rastrigin's Function	$[-100, 100]^n$	800
F11	Shifted and Rotated Rastrigin's Function	$[-100, 100]^n$	900

The results of the tests were compared with performances of CLPSO, HCLPSO, PSO, GL-PSO and GGL-PSOD. The parameter settings of this algorithms are listed in Table 3.

Table 3. Parameters settings.

Algorithm	Parameter settings
CLPSO	$w=0.9-0.4, c=1.496$
HCLPSO	$w=0.99-0.2, c_1=2.5-0.5, c_2=0.5-2.5, c=3-1.5$
PSO	$w=0.9-0.4, c_1=2.0, c_2=2.0$
GL-PSO	$w=0.7298, c=1.49618, p_m=0.01, s_g=7$
GL-PSOD	$w=0.7298, c=1.49618, p_m=0.01, s_g=7$

Both in the GL-PSOI and GL-PSOIF, the inertia weight $w = 0.6$ [13]. The acceleration coefficients used in the computations were equal $c_1=c_2=1.7$. In case of the set of benchmark functions, the population consisted of 20 particles, the dimension of the search space was 30, the maximum number of function evaluations was 300000. The search range depends on the function used as shown in Table 1. For each problem, the simulations were run 30 times. For CEC2014 functions, the population consisted of 50 particles, the dimension of the search space was $D=30$, and the maximum number of function evaluations was $D \times 10^4$. The search range was $[-100, 100]^n$. For CEC2014 functions, the algorithms were run 31 times independently.

The exemplary results of the tests are summarized in Tables 4-5.

Table 4. The comparison test results of the PSO algorithms on the benchmark functions.

Functions	Criteria	CLPSO	HCLPSO	GL-PSO	PSO	GGL-PSOD	GL-PSOI	GL-PSOIF
F1	Mean	0.00E+00(=)	0.00E+00(=)	0.00E+00(=)	3.48E-25(+)	0.00E+00(=)	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	2.08E-24	0.00E+00	0.00E+00	0.00E+00
F2	Mean	6.88E+01(+)	5.57E+00(+)	2.43E-20(+)	2.71E-11(+)	6.74E-20(+)	3.15E-22	4.52E-21
	Std	3.24E+01	4.03E+00	3.16E-20	4.29E-11	4.82E-20	2.67E-21	3.84E-20
F3	Mean	2.34E+01(+)	2.16E+00(+)	6.48E-01(+)	4.16E+01(+)	6.53E-01(+)	5.02E-01	5.16E-01
	Std	1.58E+01	4.24E+00	2.54E-01	3.92E+01	6.07E-01	5.48E-01	2.58E-01
F4	Mean	1.02E-11(+)	6.32E-12(+)	7.14E-14(+)	3.89E+01(+)	4.32E-14(+)	6.44E-15	3.50E-16
	Std	3.21E-12	8.40E-12	3.62E-14	9.22E+00	5.36E-14	5.37E-14	3.68E-15
F5	Mean	2.05E-14(+)	1.41E-12(+)	7.86E-15(+)	3.59E-13(+)	6.29E-15(+)	5.85E-15	5.32E-16
	Std	3.41E-15	4.07E-13	3.92E-15	7.91E-14	2.23E-15	2.73E-15	1.98E-15
F6	Mean	1.82E-32(+)	1.65E-32(+)	1.73E-31(+)	3.47E-02(+)	2.11E-31(+)	1.62E-32	1.57E-32
	Std	5.56E-48	5.56E-48	1.94E-32	5.89E-02	3.73E-32	5.04E-36	4.86E-34

adfa, p. 8, 2011.

© Springer-Verlag Berlin Heidelberg 2011

Table 5. The comparison test results of the PSO algorithms on the CEC2014 test suite.

Functions	Criteria	CLPSO	HCLPSO	GL-PSO	PSO	GGL-PSOD	GL-PSOI	GL-PSOIF
F7	Mean	3.24E+02(-)	4.15E+02(-)	5.96E+02(+)	8.09E+02(+)	7.12E+02(+)	4.58E+02	4.41E+02
	Std	4.85E+02	6.73E+02	3.63E+02	3.34E+02	7.29E+02	6.73E+02	1.18E+02
F8	Mean	6.93E+01(+)	3.82E+01(-)	2.76E+01(-)	1.62E+02(+)	6.27E+01(+)	5.75E+01	4.64E+01
	Std	3.15E+01	3.36E+01	6.59E+01	5.16E+01	3.49E+01	5.18E+01	2.37E+01
F9	Mean	2.08E+01(=)	2.00E+01(=)	2.05E+01(=)	2.32E+01(+)	2.00E+01(=)	2.00E+01	2.00E+01
	Std	5.37E-02	6.24E-03	3.42E-02	8.89E-02	3.27E-02	2.83E-02	2.12E-02
F10	Mean	4.07E-02(+)	2.38E-01(+)	1.95E-10(+)	2.66E+01(+)	2.43E-12(+)	2.35E-13	1.57E-13
	Std	2.19E-02	5.40E-01	7.23E-11	8.19E+00	7.68E-13	6.48E-13	1.88E-13
F11	Mean	4.20E+01(+)	4.43E+01(+)	5.84E+01(+)	7.81E+01(+)	3.57E+01(+)	2.97E+01	2.35E+01
	Std	7.17E+00	1.26E+01	2.13E+01	2.69E+01	1.49E+01	1.56E+01	1.06E+01

Table 6. The comparison test results of the PSO algorithms.

Signature	CLPSO	HCLPSO	GL-PSO	PSO	GGL-PSOD
+	8	7	8	11	8
-	1	2	1	0	1
=	2	2	2	0	2

adfa, p. 9, 2011.

© Springer-Verlag Berlin Heidelberg 2011

The exemplary charts showing the mean fitness selected functions in the following iterations for GL-PSO, GGL-PSO, CLPSO, HCLPSO, PSO, GL-PSOI and GL-PSOIF algorithms, are depicted in Figs. 1-3.

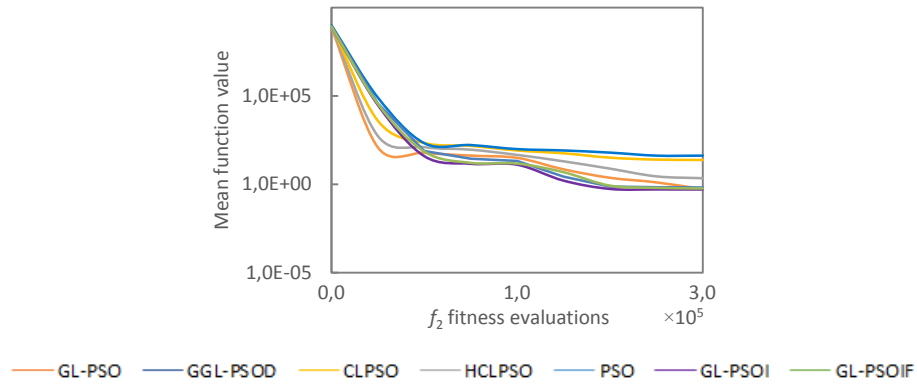


Fig. 1. Convergence performance for f_2 function.

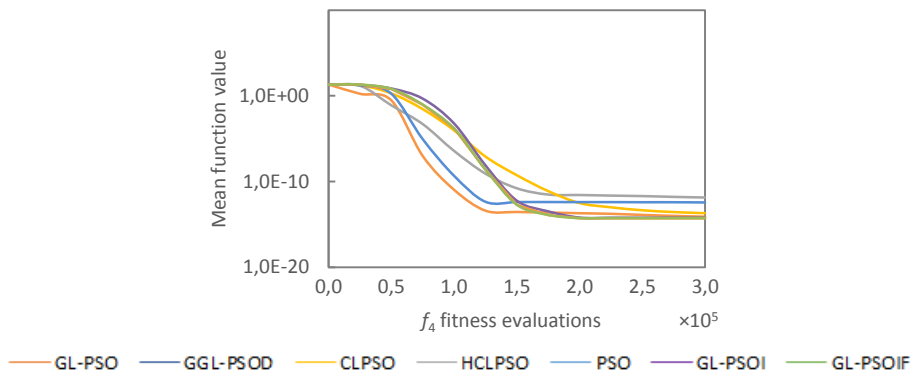


Fig. 2. Convergence performance for f_4 function.

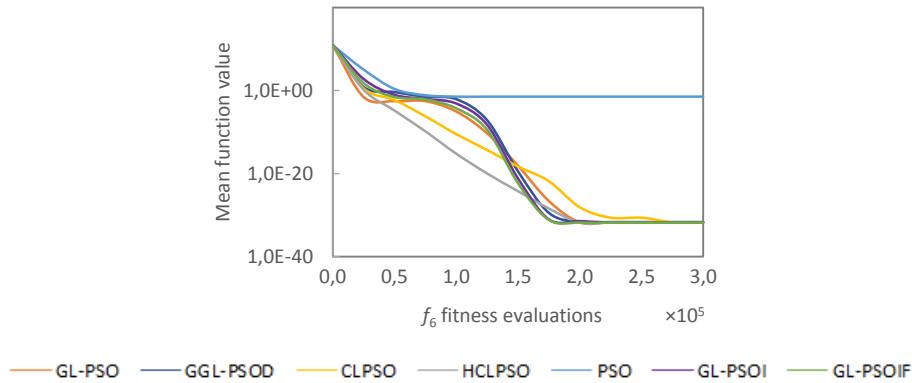


Fig. 3. Convergence performance for f_6 function.

The results of the tests confirmed that both GL-PSOI and GL-PSOIF are more effective and can achieve superior performance over the remaining tested methods. In case of unimodal functions, the GL-PSOI with interlaced ring topology obtained superior results over the ones for GL-PSOIF. For multimodal functions superior results were achieved by GL-PSOIF.

In case f_2 function, GL-PSO achieved worse results than GL-PSOI and GL-PSOIF but better than those obtained by the CLPSO, HCLPSO and PSO. For f_3 function, GL-PSOI achieved the best result. The performance of GL-PSO was worse than that obtained by GL-PSOI but superior then performance of GL-PSOIF. For unimodal f_7 function the best results were obtained by CLPSO. The outcomes achieved by GL-PSOI and GL-PSOIF were worse than results obtained by CLPSO but better than the results achieved by the remaining tested methods. For multimodal functions, the results show that (almost in all cases) GL-PSOIF exhibit the best performance.

The convergence curves presented in Figs. 1-3 indicate that both GL-PSOI and GL-PSOIF converge slower in the early stage of the optimization process than most of the compared methods. At this stage, each algorithm, except PSO, is faster. Then both algorithms accelerate and converge faster than the others.

In case of the unimodal f_2 function, both algorithms initially revealed slower convergence, which was followed by a further rapid acceleration after about 5×10^4 iterations showing superiority over the rest evaluated methods. For the unimodal f_2 function, GL-PSOIF performed a bit slower than GL-PSOI, which could be due to the introduction of flexible search operator, which did not improved the GL-PSOIF run. In case multimodal functions (Figs, 2-3), GL-PSOIF converges slowly (other methods are faster) but after about 1.3×10^5 iterations accelerates and after 2×10^5 iterations becomes the fastest.

6 Statistical test

In order to evaluate the differences between algorithms, a statistical t-test was used. A confidence level of 0,05 was selected for all statistical comparisons. Tables 4-5 shows the results of the t-test performed on the test functions. The signature '+' indicates that GL-PSOIF is significantly better than the other algorithms, '-' worse to the other algorithms, and '=' equal to the other algorithms. The rows in Table 6 named '+', '-' and '=' mean the number of times that the GL-PSOIF is better than, worse than or equal to the other algorithms. The results of the t-test indicate that proposed algorithm is significantly better than other methods with 95% confidence level in a statistically meaningful way.

7 Conclusion

In this study, a new genetic learning particle swarm optimization with interlaced ring topology and flexible local search operator (GL-PSOIF) has been proposed. To assess the impact of introduced modifications on performance of the evaluated method, first the interlaced ring topology was integrated with GL-PSO only (referred to as GL-PSOI) and then with the flexible local search operator (GL-PSOIF). The efficiency of the new strategy was tested on a set of benchmark problems and the CEC2014 test suite. The results were compared with five different variants of PSO, including GL-PSO, GGL-PSOD, PSO, CLPSO and HCLPSO. The results of the experimental trials indicated that the genetic learning particle swarm optimization with interlaced ring topology is effective for unimodal function. In case of the multimodal function, GL-PSOIF showed superior performance over the remaining tested methods.

References

1. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. IEEE International Conference on Neural Networks, pp. 1942–1948. Perth, Australia (1995).
2. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001).
3. Ignat, A., Lazar, E., Petreus, D.: Energy management for an islanded microgrid based on Particle Swarm Optimization. IEEE 24th International Symposium for Design and Technology in Electronic Packaging, SIITME 2018, pp.213–216, Romania (2018).
4. Wu, D., Gao, H.: An Adaptive Particle Swarm Optimization for Engine Parameter Optimization. Proc. Natl. Acad. Sci., India, Sect. A Phys. Sci. 88, 121-128 (2018). <https://doi.org/10.1007/s40010-016-0320-y>
5. Hu, Z., Chang, J., Zhou, Z.: PSO Scheduling Strategy for Task Load in Cloud Computing. Hunan Daxue Xuebao/Journal of Hunan University Natural Sciences 46 (8), 117-123 (2019).
6. Zhang, X., Lu, D., Zhang, X. et al. Antenna array design by a contraction adaptive particle swarm optimization algorithm. J Wireless Com Network 2019, 57 (2019). <https://doi.org/10.1186/s13638-019-1379-3>

7. Yu, M., Liang, J., Qu, B., Yue, C.: Optimization of UWB Antenna Based on Particle Swarm Optimization Algorithm. In: Li, K., Li, W., Chen, Z., Liu, Y. (eds) Computational Intelligence and Intelligent Systems. ISICA 2017. Communications in Computer and Information Science, vol. 874. Springer, Singapore (2018).
8. You, Z., Lu, C.: A heuristic fault diagnosis approach for electro-hydraulic control system based on hybrid particle swarm optimization and Levenberg–Marquardt algorithm. *J Ambient Intell Human Comput (Journal of Ambient Intelligence and Humanized Computing)* (2018). <https://doi.org/10.1007/s12652-018-0962-5>
9. Fernandes Junior, F.E., Yen, G.G.: Particle swarm optimization of deep neural networks architectures for image classification. *Swarm and Evolutionary Computation* 49, 62–74 (2019).
10. Borowska, B.: An improved CPSO algorithm. *International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT)*, pp. 1–3, IEEE, Lviv (2016). doi: 10.1109/STC-CSIT.2016.7589854
11. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. Congress on evolutionary computation, pp. 1945–1949, Washington D.C., USA (1999).
12. Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, Proc. ICEC, pp. 1951–1957, Washington, DC, (1999).
13. Trelea, I.C.: The particle swarm optimization algorithm : convergence analysis and parameter selection. *Information Processing Letters* 85, 317–325 (2003).
14. Borowska, B.: Nonlinear inertia weight in particle swarm optimization. *International Scientific and Technical Conference, Computer Science and Information Technologies, CSIT 2017*, pp. 296–299, Lviv, Ukraine (2017).
15. Borowska, B.: Influence of social coefficient on swarm motion. *Lecture Notes in Computer Science 11508, ICAISC'19*, 412–420 (2019).
16. Ratnaveera, A., Halgamuge, S.K., Watson, H.C.: Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation* 8 (3), 240–255 (2004).
17. Lu, H., Chen, W.: Self-adaptive velocity particle swarm optimization for solving constrained optimization problems. *J. Glob.Optim.* 41, 427–445 (2008).
18. Borowska, B.: Novel algorithms of particle swarm optimisation with decision criteria. *Journal of Experimental & Theoretical Artificial Intelligence* 30 (5), 615–635 (2018) DOI: 10.1080/0952813X.2018.1467491
19. Mahmoud, K.R., El-Adawy M., Ibrahim, S.M.M.: A comparison between circular and hexagonal array geometries for smart antenna systems using particle swarm optimization algorithm. *Progress in electromagnetics Research* 72, 75–90, (2007).
20. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. Proc. IEEE Congr. Evol. Comput., vol. 2, pp. 1671–1676, Honolulu, HI, USA, (2002).
21. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: simpler, maybe better, *IEEE Trans. Evol. Comput.* 8, 204–210. (2004).
22. Gong, Y.J., Li, J.J., Zhou, Y.C., Li, Y., Chung, H.S.H., Shi, Y.H., et al.: Genetic learning particle swarm optimization, *IEEE Trans. Cybern.* 46 (10), 2277–2290 (2016).
23. Lin, A., Sun, W., Yu, H., Wu, G., Tang, H.: Global genetic learning particle swarm optimization with diversity enhanced by ring topology. *Swarm and Evolutionary Computation* 44, 571–583 (2019).
24. Liang, J.J., Suganthan, P.N.: Dynamic multi-swarm particle swarm optimizer, Proc. Swarm Intell. Symp., pp. 124–129 (2005).
25. Chen, Y., Li, L., Peng, H., Xiao, J., Wu, Q.T.: Dynamic multi-swarm differential learning particle swarm optimizer, *Swarm Evol. Comput.* 39 (2017).

26. Wang, L., Yang, B., Chen, Y.H.: Improving particle swarm optimization using multilayer searching strategy, *Inf. Sci.* 274, 70–94 (2014).
27. Ye, W., Feng, W., Fan, S.: A novel multi-swarm particle swarm optimization with dynamic learning strategy. *Applied Soft Computing* 61, 832–843 (2017).
28. 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), 281–295 (2006).
29. Lin, A., Sun, W., Yu, H., Wu, G., Tang, H.: Adaptive comprehensive learning particle swarm optimization with cooperative archive. *Applied Soft Computing Journal* 77, 533–546 (2019).
30. Cheng, R., Jin, Y.: A social learning particle swarm optimization algorithm for scalable optimization, *Inf. Sci.* 291, 43–60 (2015).
31. Holden, N., Freitas, A.A.: A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data. *Proc. IEEE SIS*, pp.100–107 (2005).
32. Li, L., Wang, L., Liu, L.: An effective hybrid PSOSA strategy for optimization and its application to parameter estimation. *Applied Mathematics and Computation* 179, 135–146 (2006).
33. Shieh, H.L., Kuo, C.C., Chiang, C.M.: Modified particle swarm optimization algorithm with simulated annealing behavior and its numerical verification, *Appl. Math. Comput.* 218, 4365–4383 (2011).
34. Tian, D. Shi, Z.: MPSO: modified particle swarm optimization and its applications, *Swarm Evol. Comput.* 41, 49–68 (2018).
35. Chen, X., Tianfield, H., Mei, C.L., Du, W.L., Liu, G.H.: Biogeography-based learning particle swarm optimization, *Soft Comput.* 21, 7519–7541 (2017).
36. Bouyer, A., Hatamlou, A.: An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms, *Appl. Soft Comput.* 67, 172–182 (2018).
37. Duraj A., Chomatek L.: Outlier Detection Using the Multiobjective Genetic Algorithm, *Journal of Applied Computer Science* 25 (2), 29–42 (2017).
38. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, Technical Report, Nanyang Technological University, Singapore (2013).
39. Lynn, N., Suganthan, P.N.: Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, *Swarm and Evolutionary Computation* 24, 11–24 (2015).