PARTICLE SWARM OPTIMIZATION WITH DISTANCE BASED REPULSIVITY

Michal Pluhacek[1], Ivan Zelinka[2], Roman Senkerik[1], Adam Viktorin[1], Tomas Kadavy[1]

[1]Tomas Bata University in Zlin
Faculty of Applied Informatics
Nad Stranemi 4511, 760 05 Zlin
Czech Republic
pluhacek@utb.cz, senkerik@utb.cz, aviktorin@utb.cz, kadavy@utb.cz

[2]Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
17. listopadu 15,708 33 Ostrava-Poruba
Czech Republic
ivan.zelinka@vsb.cz

Abstract: *In this study, we propose a repulsive mechanism for the Particle Swarm Optimization algorithm that improves its performance on multi-modal problems. The repulsive mechanism is further extended with a distance-based modification. The results are presented and tested for statistical significance. We discuss the observations and propose further directions for the research.*

Keywords: *Swarm Intelligence, Particle Swarm Optimization, Repulsive, Distance based*

1 Introduction

The Particle Swarm Optimization (PSO) [1-4] is one of the most popular metaheuristics for global optimization if not the most popular (according to the number of research papers published every year). The method is widely used in all areas of industrial optimization and remains in the center of interest of the research community [5,6].

One of the well-known weakness of PSO is the inclination to premature convergence into local optima. In [7] a repulsive mechanism was proposed to address this issue and serves as an inspiration for this work. In this initial study, the core PSO formula is altered by introducing a particle-to-particle repulsivity and a distance-based repulsivity multiplier. The aim is to find a balanced method for both smooth unimodal and complex multi-modal fitness landscapes. The experimental part utilizes four well-known benchmark functions.

The main aim of this initial study is not to present a developed highly competitive method but to highlight the usefulness of the repulsive mechanic and serve as a background for future research in this direction.

The rest of the paper is structured as follows: In section 2, the PSO algorithm is described. In section 3, the proposed modification is detailed. The experimental details are given in section 4. The results are presented in section 5, followed by a discussion and the conclusion.

2 Particle Swarm Optimization (PSO)

The original PSO [1] takes the inspiration from the flocking behavior of birds. In the start, a population (swarm) of candidate solutions (particles) of the optimization problem (defined by a cost function) is randomly generated. Each particle is evaluated (assigned a quality quantification using the cost function). Next, the particles simulate a bird flight over the fitness landscape. The knowledge of the global best-found solution (typically noted *gBest*) is shared among the particles in the swarm. Furthermore, each particle has the knowledge of its own (personal) best-found solution (noted *pBest*). Last important part of the algorithm is the velocity of each particle that is taken into account during the calculation of the particle movement. The new position of each particle is then given by (1), where $\vec{x}_i^{t+1}$ is the new particle position; $\vec{x}_i^t$ refers to the current particle position and $\vec{v}_i^{t+1}$is the new velocity of the particle.

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} \tag{1}$$

To calculate the new velocity, the distance from *pBest* and *gBest* is taken into account alongside with current velocity (2).

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot Rand \cdot (pBest_{ij} - x_{ij}^{t}) + c_2 \cdot Rand \cdot (gBest_j - x_{ij}^{t}) \quad (2)$$

Where:

v_{ij}^{t+1}	– New velocity of the ith particle in iteration $t+1$ (component j of the dimension *D*).
w	– Inertia weight value. v_{ij}^{t+1}– Current velocity of the ith particle, iteration t (component j of the dimension *D*).
c_1, c_2	– Acceleration constants.
$pBest_{ij}$	– Local (personal) best solution found by the ith particle. (component j of the dimension *D*).
$gBest_j$	– Best solution found in a population. (component j of the dimension *D*).
x_{ij}^{t}	– Current position of the ith particle (component j of the dimension *D*) in iteration *t*.
$Rand$	– Pseudo random number, interval (0, 1).

After the movement, the particle evaluates the quality of its new position and compares it with its personal best solution (*pBest*). If a better value was discovered, the *pBest* is updated. Similarly, if the new best solution in the neighborhood (swarm or sub-swarm) was discovered, the *gBest* is updated.

There are two models for *pBest*/*gBest* updating, the Asynchronous and Synchronous updating [8]. In the Synchronous model, the *pBest* and *gBest* are updated in one moment for all particles (therefore only one update of *gBest* per iteration is performed). In the Asynchronous model, the *pBest* and *gBest* are updated immediately after the better value is discovered (therefore, multiple updates of *gBest* per iteration are possible).

3 Proposed Modification

In the original PSO, the exploration phase often ends prematurely, and the algorithm starts exploiting the current sub-optima almost instantly. This leads to a poor performance when facing complex multi-modal problems. To prevent the swarm from instant convergence, a partial repulsivity is proposed in this work. The swarm is indexed into a ring topology; therefore, it is easy to assign each particle a single neighbor with indexes higher by one. Given the ring topology, the last particle in the population is neighboring the first particle in the population. The velocity calculation formula is altered to the form given by (3) and (4).

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot Rand_1 \cdot \left(pBest_{ij} - x_{ij}^{t}\right) + c_2 \cdot Rand_2 \cdot \left(gBest_j - x_{ij}^{t}\right) - c_3 \cdot Rand_3 \cdot \left(x_{kj}^{t} - x_{ij}^{t}\right) \quad (3)$$

Where:

c_3	– repulsive constant
$Rand_3$	– Pseudo-random number, uniform distribution, interval (0, 1).
k	– index of neighboring particle in the population (4)

$$k = (i \; mod \; NP) + 1 \quad (4)$$

Where:

i	– index of the active particle, $i \in \langle 1; NP \rangle$
NP	– population size

Each particle is partially repulsed from its right-hand side neighbor in the ring topology. This unique repulsivity on single-particle level helps slow down the convergence of the swarm but does not prevent the convergence completely. As no two particles share the same point of repulsivity, the natural movement of the swarm is maintained. Using this method, the exploration phase is prolonged, while the exploitation phase is not abandoned. After extensive testing, the value of c_3 has been set to 0.15. For this value, the algorithm seems to achieve the best performance.

3.1 Distance-based Approach

In the above-presented design, the repulsive force is affecting the particles regardless of their mutual distance, and similar force is applied in each dimension. However, it seems more natural that the repulsive force should increase with the particles near each other. For this reason, a distance-based repulsivity multiplier is introduced in (5).

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot Rand_1 \cdot \left(pBest_{ij} - x_{ij}^{t}\right) + c_2 \cdot Rand_2 \cdot \left(gBest_j - x_{ij}^{t}\right) - c_3 \cdot rep_j \cdot Rand_3 \cdot \left(x_{kj}^{t} - x_{ij}^{t}\right) \quad (5)$$

Where: rep_j – Distance based repulsivity multiplier (component j of the dimension *D*).

The value of the distance-based multiplier is given by (6). The formula was derived after extensive testing.

$$rep_j = -\left(1/\left(1 + e^{0.01\left(-\left|x_{kj}^t - x_{ij}^t\right| + (\text{range}_j/4)\right)}\right)\right) \tag{6}$$

Where: range_j – objective function range (component j of the dimension D).

An exemplary depiction of the value of Distance-based repulsivity multiplier is presented in Fig. 1.

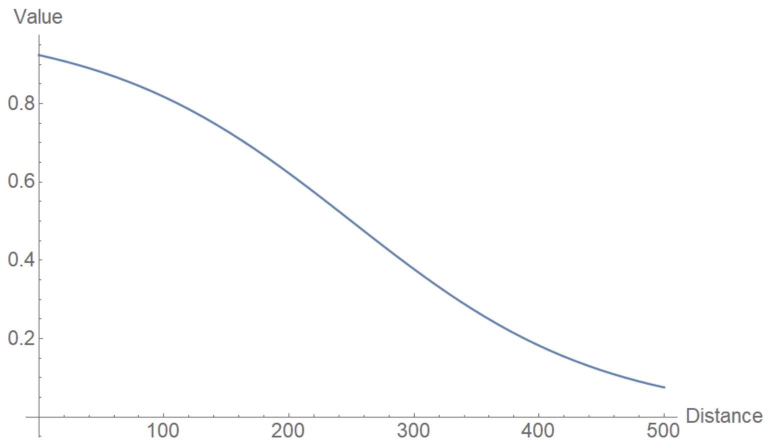

Figure 1: Distance-based repulsivity multiplier values (example for Schwefel function, range = 1000)

4 Experiment

In the experimental part, the performance of both proposed modifications was tested using well-known unimodal and multimodal functions [9] in several different dimensional settings. To provide a fair comparison, the newly proposed modifications are compared with both asynchronous and synchronous updating PSOs. The control parameters were set as follows (in accordance with common values and recommendations in [6]):

NP: 30; max. CFE: 30 000; dim = 10, 30, 50; w = 0.7298; c_1, c_2 = 1.49618; c_3 = 0.15;

Following set of four common test functions was used:
The Sphere function is given by (7).

$$f(x) = \sum_{i=1}^{dim} x_i^2 \tag{7}$$

Function minimum:
Position for E_n: $(x_1, x_2, \dots, x_n)$ = (0, 0, …, 0); Value for E_n: $f(x) = 0$

The Rosenbrock's function is given by (8).

$$f(x) = \sum_{i=1}^{dim-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \tag{8}$$

Function minimum:
Position for E_n: $(x_1, x_2, \dots, x_n)$ = (1, 1, …, 1); Value for E_n: $f(x) = 0$

The Rastrigin's function is given by (9).

$$f(x) = 10\,dim + \sum_{i=1}^{dim} x_i^2 - 10\cos(2\pi x_i) \tag{9}$$

Function minimum:
Position for E_n: $(x_1, x_2, \dots, x_n)$ = (0, 0, …, 0); Value for E_n: $f(x) = 0$

The Schwefel's function is given by (10)

$$f(x) = \sum_{i=1}^{dim} -x_i \sin(\sqrt{|x|}) \quad (10)$$

Function minimum:
Position for E$_n$: $(x_1, x_2, \dots, x_n)$ = (420.969, 420.969, …, 420.969); Value for E$_n$: $f(x)$ = 0

4.1 Results

In Table 1, the mean results of all compared algorithms are presented. Further, the Wilcoxon rank sum test with a level of significance alpha 0.05 was used to test all pairs of algorithms. The winner is identified if it managed to outperform all remaining algorithms with statistical significance. Otherwise, the winner is not designated.

In subsequent Table 2 similar Wilcoxon test results are given for selected combinations of algorithms. The notation is as follows: A1 – Asynchronous PSO, A2 – Synchronous PSO, A3 – PSO Repuls, A4 – PSO Repuls 2 (with distance-based multiplier).

In Fig. 2, the rankings according to Friedman rank sum test are given alongside with the Nemenyi posthoc test critical distance line. Finally, as an example, the mean history of *gBest* value comparisons for dim = 30 are given in Fig. 3 – 6.

Table 1: Mean results overview, Wilcoxon test winners noted

function	*PSO Async.*	*PSO Sync.*	*PSO Repuls.*	*PSO Repuls. 2*	*winner*
			dim = **10**		
Sphere	0.00E+00	0.00E+00	2.94E-28	0.00E+00	-
Rosenbrock	2.83E+01	1.92E+01	2.24E+01	1.90E+01	-
Rastrigin	7.23E+00	7.96E+00	4.59E+00	6.43E+00	PSO Repuls.
Schwefel	9.44E+02	9.68E+02	9.72E+01	4.96E+02	PSO Repuls.
			dim = **30**		
Sphere	3.42E-15	1.01E-12	3.57E-09	1.02E-11	PSO Async.
Rosenbrock	1.07E+02	7.49E+01	1.46E+02	8.73E+01	-
Rastrigin	1.19E+02	1.28E+02	6.84E+01	9.66E+01	PSO Repuls.
Schwefel	4.48E+03	4.76E+03	2.03E+03	2.84E+03	PSO Repuls.
			dim = **50**		
Sphere	3.62E-05	1.32E-02	2.37E-03	4.83E-04	PSO Async.
Rosenbrock	1.84E+02	3.38E+02	5.47E+02	2.85E+02	PSO Async.
Rastrigin	4.05E+02	4.93E+02	2.35E+02	2.83E+02	PSO Repuls.
Schwefel	8.62E+03	8.61E+03	4.53E+03	5.48E+03	PSO Repuls.

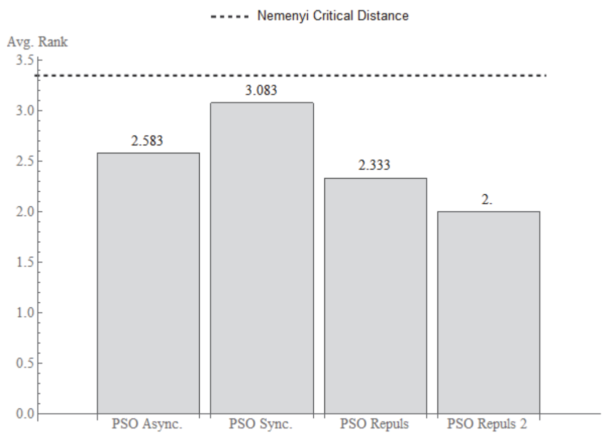

Figure 2: The Friedman ranks with the Nemenyi posthoc test

Table 2: Wilcoxon tests winners

function	A1 / A2 / A3	A1 / A2 / A4	A3 / A4
		dim = **10**	
Sphere	-	-	A4
Rosenbrock	-	-	-
Rastrigin	A3	-	A3
Schwefel	A3	A4	A3
		dim = **30**	
Sphere	A1	A1	A4
Rosenbrock	-	-	-
Rastrigin	A3	A4	A3
Schwefel	A3	A4	A3
		dim = **50**	
Sphere	A1	A1	A4
Rosenbrock	A1	A1	A4
Rastrigin	A3	A4	A3
Schwefel	A3	A4	A3

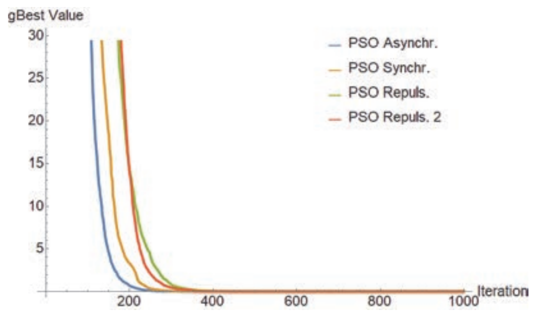

Figure 3: Mean *gBest* history – Sphere function dim = 30

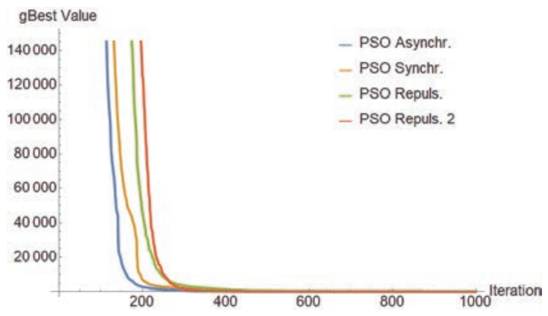

Figure 4: Mean *gBest* history – Rosenbrock function dim = 30

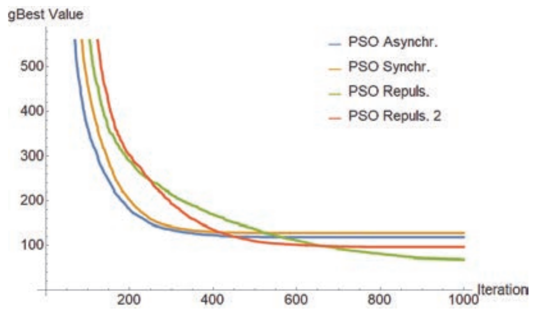

Figure 5: Mean gBest history – Rastrigin function dim = 30

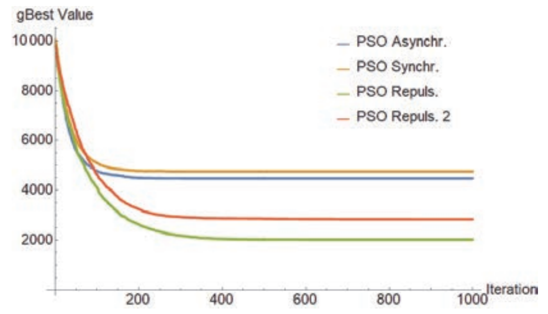

Figure 6: Mean gBest history – Schwefel function dim = 30

5 Discussion

According to the above-presented results, the introduction of particle-to-particle repulsivity leads to significant improvement of the performance of the algorithm on multi-modal problems. However, the performance of smoother landscapes is significantly worse. According to Fig. 2, the distance-based modification presents the best overall performance. It seems that the variant noted as PSO Repuls. 2 is balanced for use on any shape of the fitness landscape. However, given that all ranks are within the critical distance of post-hoc test, it is necessary to further improve the method to achieve results of statistical significance across the whole benchmark set.

The convergence speed seems comparable (Fig. 3 -6). However, both proposed modifications seem to avoid local optima better than either asynchronous or synchronous PSO.

6 Conclusion

In this work, a modification of the Particle Swarm Optimization core formula is proposed. The goal is to improve the performance of the algorithm on complex multimodal problems and higher dimensions. According to the evidence, it seems the proposed method is capable of better avoiding the local optima on the multi-modal landscape. However, a more balanced performance is achievable when a distance based repulsivity multiplier is introduced. This initial design shows very promising performance. However it needs a further enhancement that will be the main focus of the future research.

Acknowledgement: This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project no. LO1303 (MSMT-7778/2014), further by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the Projects no. IGA/CebiaTech/2018/003. This work is also based upon support by COST (European Cooperation in Science & Technology) under Action CA15140, Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO), and Action IC1406, High-Performance Modelling and Simulation for Big Data Applications (cHiPSet). The work was further supported by resources of A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin (ailab.fai.utb.cz).

References

[1] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)

[2] Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 69–73 (1998)

[3] Kennedy, J.: The particle swarm: social adaptation of knowledge. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 303–308 (1997)

[4] Nickabadi, A., Ebadzadeh, M.M., Safabakhsh R.: A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing **11**(4), 3658–3670 (2011), ISSN 1568-4946

[5] Eberhart, R.C., Shi Y.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, San Diego, USA, pp. 84–88 (2000)

[6] Van Den Bergh, F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. Information Sciences **176** (8), 937–971 (2006)

[7] Riget, J., Vesterstrøm, J.S.: A diversity-guided particle swarm optimizer-the ARPSO. Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark (2002)

[8] Engelbrecht, A.P.: Particle Swarm Optimization: Iteration Strategies Revisited. In: BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, Ipojuca, pp. 119–123 (2013)

[9] Dieterich, J.M., Hartke B.: Empirical review of standard benchmark functions using evolutionary global optimization. arXiv preprint arXiv:1207.4318 (2012)