A Study of Particle Swarm Optimization with Considering More Local Best Particles

Yen-Ching Chang
Department of Medical Informatics, Chung Shan Medical University and Department of Medical Imaging, Chung Shan Medical University Hospital
Taichung, Taiwan
e-mail: nicholas@csmu.edu.tw

Li-Chun Lai
Bachelor Program in Robotics
National Pingtung University of Education
Pingtung, Taiwan
e-mail: lclai@mail.npue.edu.tw

Chin-Chen Chueh, Yongxuan Xu, Cheng-Hsueh Hsieh
Department of Medical Informatics
Chung Shan Medical University
Taichung, Taiwan
e-mail: paster0823@gmail.com, itismynamerobert@yahoo.com.tw, cvicivity@gmail.com

***Abstract*—The velocity updating formula of the standard particle swarm optimization (PSO) only considers two particles: the local best particle and the global best particle. The global best particle can be viewed as the optimal one of all local best particles. In order to improve the optimizing performance and exploit all existing resources as fully as possible, we further study other local best particles how to affect the results of optimization in this paper. Experimental results show that a suitable selection of the number of local best particles will result in higher performance.**

Keywords-optimization; particle swarm; algorithm; particle swarm optimization

I. INTRODUCTION

Particle swarm optimization (PSO), first developed by Kennedy and Eberhart [1], is a useful optimization technique inspired by swarm intelligence. In their paper, a concept for optimizing nonlinear functions by using particle swarm methodology was introduced. Since then, PSO has come a long way and gone through many changes as well as variants. One of these courses includes a modified particle swarm optimizer provided by Shi and Eberhart [2]. In their paper, a new parameter, called inertia weight, was introduced into the original particle swarm optimization. Some simulations were illustrated how the new parameter influence the results. After that study, Shi and Eberhart provided some guidelines for selecting inertia weight and maximum velocity [3]. In addition, both authors proposed an empirical study of particle swarm optimization [4], which provided a linearly decreasing inertia weight with beginning at 0.9 and ending at 0.4 as well as setting two positive constants as 2. The first phase and category of PSO mainly centers on the influence of inertia weight, parameter selection, the effect of velocity limit, etc.

In the second category of PSO, the stability of PSO and its related concepts were studied and discussed. Clerc and Kennedy explored how an individual particle worked in the particle swarm algorithm [5]. They applied the constriction constant to control over the dynamical characteristics of the particle swarm such as a tendency for exploration and exploitation. Trelea analyzed the dynamical behavior and convergence of the simplified (deterministic) PSO algorithm from the dynamic system theory and further provided a guideline of graphical parameter selection [6]. Jiang applied the stochastic process theory to analyze the standard PSO algorithm determined by three parameters including inertia weight and two coefficients for the local best particle and the global best particle [7].

In the third category of PSO, some variants of swarm were introduced in order to obtain better performance of optimization; these include bare bone particle swarm [8-10], Gaussian swarm [11-13], some versions of jump or mutation with or without new search strategies [9-10,12,14-15], and different topologies of particle-grouping [16-17]. For more detailed information, Poli et al. provided an overview of PSO [18].

The rest of the paper is organized as follows. Section II introduces the standard PSO. In Section III, we propose the concept of PSO with considering more local best particles. Simulation results for five benchmark nonlinear functions are provided and discussed in Section IV. Finally, the paper is summarized in Section V.

II. STANDARD PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population-based algorithm [1]. At first, the inertia weight didn't be introduced into the particle swarm to optimize the objective function. For convenience and consistence, the PSO with an inertia weight is called the standard PSO, or simply PSO, though some authors might name it the original or canonical PSO. The PSO is initialized by an array of particles with random positions and velocities on *d* dimensions in the search space. Each particle, expressed by a coordinate, denotes a potential optimal solution of an objective function. It moves through the search space to find the optimal solution according to the following two iterative equations for velocity and position:

$$\boldsymbol{v}_i^{k+1} = w\boldsymbol{v}_i^k + c_1\boldsymbol{r}_1 \otimes \left(\boldsymbol{p}_i^k - \boldsymbol{x}_i^k\right) + c_2\boldsymbol{r}_2 \otimes \left(\boldsymbol{p}_g^k - \boldsymbol{x}_i^k\right) \tag{1}$$

and

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \boldsymbol{v}_i^{k+1}, \tag{2}$$

where w is a controllable parameter called inertia weight; c_1 and c_2 are two positive constants or coefficients called cognitive and social parameters, respectively; $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are two vectors of random numbers uniformly distributed in the range $(0,1)$; the row vector $\boldsymbol{x}_i^k = (x_{i1}, x_{i2}, \ldots, x_{id})$ with d dimensions stands for the current (kth iteration) position of the ith particle in the Cartesian coordinate system; $\boldsymbol{v}_i^k = (v_{i1}, v_{i2}, \ldots, v_{id})$, the current velocity of the ith particle; $\boldsymbol{p}_i^k = (p_{i1}, p_{i2}, \ldots, p_{id})$, the present (among k iterations) best position particle i explores after k steps or simply the local best position of the ith particle; $\boldsymbol{p}_g^k = (p_{g1}, p_{g2}, \ldots, p_{gd})$, the optimal one of all local best positions or simply the global best position; finally, $\otimes$ denotes a element-by-element multiplication for vectors or matrix.

III. Two New Types of Variants of PSO

In the standard PSO, only two particles or positions are considered in the updating formula of velocity: one is the local best position of each particle, $\boldsymbol{p}_i^k$, and the other is the global best position, $\boldsymbol{p}_g^k$. Since the global best position, the optimal position of all local best particles, contributes to finding a better solution, the second optimal position of all local best particles should also be useful for exploring in the search space. In order to verify our speculation, we take the first mth local best particles into consideration in this paper. The iterative equation of velocity for the first new variant of PSO can be described as follows, called the PSO with more local best particles of type 1:

$$\begin{aligned}\boldsymbol{v}_i^{k+1} = {} & w\boldsymbol{v}_i^k + c_1\boldsymbol{r}_1 \otimes \left(\boldsymbol{p}_i^k - \boldsymbol{x}_i^k\right) + c_2\boldsymbol{r}_2 \otimes \left(\boldsymbol{p}_{g1}^k - \boldsymbol{x}_i^k\right) \\ & + c_3\boldsymbol{r}_3 \otimes \left(\boldsymbol{p}_{g2}^k - \boldsymbol{x}_i^k\right) + \cdots + c_{m+1}\boldsymbol{r}_{m+1} \otimes \left(\boldsymbol{p}_{gm}^k - \boldsymbol{x}_i^k\right),\end{aligned} \tag{3}$$

where c_1 is called the cognitive parameter and c_2 to c_{m+1} are still called social parameters; $\boldsymbol{p}_{g1}^k$ denotes the optimal position of all local best particles, equal to $\boldsymbol{p}_g^k$ for the standard PSO, $\boldsymbol{p}_{g2}^k$ the second optimal position of all local best particles, and finally, $\boldsymbol{p}_{gm}^k$ the mth optimal position of all local best particles.

On the other hand, we would like to further examine how the results are when the cognitive part is omitted. The iterative equation of velocity for the second new variant of PSO can be described as follows, called the PSO with more local best particles of type 2:

$$\begin{aligned}\boldsymbol{v}_i^{k+1} = {} & w\boldsymbol{v}_i^k + c_1\boldsymbol{r}_1 \otimes \left(\boldsymbol{p}_{g1}^k - \boldsymbol{x}_i^k\right) + c_2\boldsymbol{r}_2 \otimes \left(\boldsymbol{p}_{g2}^k - \boldsymbol{x}_i^k\right) \\ & + \cdots + c_m\boldsymbol{r}_m \otimes \left(\boldsymbol{p}_{gm}^k - \boldsymbol{x}_i^k\right).\end{aligned} \tag{4}$$

TABLE I. Comparisons for Considering Different Local Best Particles with the Cognitive Part (Type 1)

m	**sphere**	**Rosenbrock**	**Rastrigin**	**Griewank**	**Schaffer's f6**
1	2.07E-92	2.94E+01	3.90E+01	1.75E-02	0.00E+00
2	3.75E-103	1.69E+01	3.12E+01	1.50E-02	0.00E+00
3	1.39E-88	5.05E+00	2.89E+01	1.48E-02	0.00E+00
4	4.51E-80	9.53E-01	2.44E+01	9.61E-03	0.00E+00
5	2.53E-89	4.29E+00	2.32E+01	1.14E-02	0.00E+00
6	9.01E-68	6.97E+00	2.03E+01	5.42E-03	0.00E+00
7	1.07E-53	1.79E+01	2.24E+01	9.23E-03	4.86E-04
8	1.48E-73	2.06E+01	1.63E+01	2.71E-03	0.00E+00
9	5.97E-59	2.76E+01	1.63E+01	2.71E-03	4.86E-04
10	5.18E-41	2.78E+01	1.60E+01	1.73E-03	4.86E-04
11	8.63E-40	2.88E+01	1.46E+01	4.06E-03	0.00E+00
12	3.71E-45	3.64E+01	1.21E+01	2.59E-03	4.86E-04
13	2.65E-37	3.38E+01	1.21E+01	3.57E-03	9.72E-04
14	8.34E-20	3.65E+01	1.14E+01	2.83E-03	5.41E-04
15	2.59E-20	3.43E+01	1.00E+01	2.46E-03	7.77E-04
16	1.76E-20	3.20E+01	1.03E+01	4.06E-03	4.97E-04
17	2.99E-14	3.59E+01	9.95E+00	2.59E-03	6.99E-04
18	3.36E-14	2.82E+01	9.60E+00	7.40E-04	1.64E-03
19	1.53E-07	3.41E+01	1.03E+01	1.23E-03	5.82E-04
20	2.12E-09	2.74E+01	9.70E+00	4.65E-04	2.45E-03
21	4.24E-14	3.60E+01	9.05E+00	1.89E-03	1.19E-03
22	1.61E-08	3.08E+01	8.90E+00	7.40E-04	7.46E-04
23	2.82E-06	3.92E+01	8.16E+00	9.84E-04	1.09E-03
24	1.14E-06	3.57E+01	8.41E+00	1.41E-03	1.14E-03
25	2.38E-07	3.49E+01	8.61E+00	6.83E-04	9.42E-04
26	5.94E-06	4.81E+01	8.41E+00	1.47E-03	1.58E-03
27	2.01E-05	3.84E+01	8.76E+00	8.95E-03	2.43E-03
28	3.63E-06	3.50E+01	6.87E+00	2.84E-03	1.58E-03
29	1.64E-06	2.89E+01	6.65E+00	3.24E-03	3.69E-03
30	3.73E-05	6.06E+01	6.37E+00	2.66E-03	2.12E-03
min	3.75E-103	9.53E-01	6.37E+00	4.65E-04	0.00E+00
max	3.73E-05	6.06E+01	3.90E+01	1.75E-02	3.69E-03

IV. Simulation

A. Benchmark Functions

The two newly proposed variants of PSO were applied in the optimization of five benchmark functions, used by Trelea [6]. They are sphere, Rosenbrock, Rastrigin, Griewank, and Schaffer's f6, respectively. The formula, the number of dimensions, the range of each dimension, and the optimum of each function can refer to [6].

B. Experimental Settings

The inertia weight is changed by a linear decrease from 0.9 to 0.4 over the course of each run. Usually, c_1 is chosen as 2 and c_2 as 2, and the sum is 4. Based on this idea, if we consider m local best particles, then constants c_1 to c_{m+1} are equal to $4/(m+1)$ for type 1, and $4/m$ for type 2. In order to study a different number of local best particles how to affect the performance of optimization, m was executed from 1 to n, the number of particles.

The number of particles (n) or the size of population was set to be 30. Each experiment was run 20 times and terminated after the maximum number of iterations (15,000) was executed. Initial positions were randomly chosen in the range of each dimension; each initial velocity was all set to be 0. During the optimizing process, the particles were limited to the range of each dimension and the velocity of each particle was restricted to 0.2 times the range of each dimension. As for particle-grouping, a fully connected topology was adopted, in which all particles are in the neighborhood of each other.

C. Experimental Results and Discussion

We simulated the results using (3) and (4), also simply called type 1 with the cognitive part and type 2 without the cognitive part, and listed them in Table I and II, respectively. The field name of the first column, m, stands for how many local best particles are included and the field names of the second to sixth columns for the names of functions studied.

The value of each cell in column 2 to 6 stands for the average of 20 (runs) function error values. Here, the function error value means the absolute difference between the existing optimal function value (maybe the minimum or maximum of each function) and best function value found after the maximum number of iteration, 15,000. The values in bold and black stands for the minimum of function error values on 30 (n) possibilities of m, and the values in bold and red for the maximum of each function error values.

1) Type 1 with the Cognitive Part: In this case, the results of type 1 with $m=1$ is exactly the same as the ones of the standard PSO. Obviously, the results of type 1 with $m=2$ are all better than the ones of the standard PSO. The results of type 1 with m equla to 3 to 5 are almost better than the ones of the standard PSO except at the sphere function. Even so, the results are also considerably acceptable. Therefore, our newly proposed PSO of type 1 is a promising idea to improve the performance of optimization.

2) Type 2 without the Cognitive Part: We find that the results of type 2 with m equal to 4 to 6 are all better than the results of type 1 with $m=1$, exactly the same as the ones of the standard PSO, and the results of type 2 with $m=3$ are almost better than the ones of the standard PSO except at function Griewank, but only a slight difference. Therefore, our newly proposed PSO of type 2 also provides a hopeful concept to improve the performance of optimization.

TABLE II. COMPARISONS FOR CONSIDERING DIFFERENT LOCAL BEST PARTICLES WITHOUT THE COGNITIVE PART (TYPE 2)

m	**sphere**	**Rosenbrock**	**Rastrigin**	**Griewank**	**Schaffer's f6**
1	1.31E-02	4.73E+03	3.26E+01	5.99E-02	0.00E+00
2	1.64E-54	4.79E+03	3.61E+01	2.51E-02	0.00E+00
3	2.61E-187	3.99E-01	3.27E+01	1.85E-02	0.00E+00
4	1.45E-200	3.99E-01	2.60E+01	1.73E-02	0.00E+00
5	1.50E-169	9.38E-01	2.25E+01	1.70E-02	0.00E+00
6	1.10E-132	4.95E+00	1.96E+01	7.51E-03	0.00E+00
7	1.13E-97	1.00E+01	1.84E+01	2.96E-03	4.86E-04
8	5.69E-92	1.38E+01	1.80E+01	5.42E-03	0.00E+00
9	1.84E-109	1.77E+01	1.66E+01	4.93E-03	0.00E+00
10	6.53E-88	2.11E+01	1.58E+01	5.18E-03	9.72E-04
11	1.06E-99	2.53E+01	1.40E+01	2.22E-03	9.72E-04
12	2.37E-78	2.60E+01	1.41E+01	2.59E-03	1.08E-04
13	6.65E-33	2.44E+01	1.19E+01	8.63E-04	7.71E-04
14	1.18E-90	3.95E+01	1.19E+01	1.48E-03	2.87E-05
15	4.88E-62	3.66E+01	1.20E+01	0.00E+00	4.86E-04
16	1.65E-45	3.15E+01	1.11E+01	0.00E+00	0.00E+00
17	4.81E-36	2.99E+01	1.07E+01	4.93E-04	2.71E-04
18	1.84E-25	3.66E+01	1.15E+01	7.40E-04	1.65E-03
19	2.21E-40	2.67E+01	1.01E+01	7.40E-04	1.14E-03
20	2.04E-18	3.41E+01	1.06E+01	2.78E-17	2.30E-03
21	2.41E-09	2.99E+01	1.00E+01	9.86E-04	8.48E-04
22	2.62E-15	2.98E+01	7.81E+00	2.78E-09	2.26E-03
23	3.94E-08	3.15E+01	7.46E+00	4.93E-04	1.81E-03
24	4.33E-12	3.03E+01	9.20E+00	1.24E-12	1.17E-03
25	8.73E-08	3.11E+01	7.96E+00	2.88E-09	2.87E-03
26	9.36E-07	3.05E+01	7.61E+00	1.36E-05	2.63E-03
27	9.56E-06	2.78E+01	7.76E+00	5.16E-06	1.70E-03
28	2.61E-11	3.89E+01	7.16E+00	6.32E-04	1.03E-03
29	2.29E-07	2.77E+01	7.21E+00	1.03E-03	2.51E-03
30	4.97E-10	3.10E+01	6.52E+00	4.80E-04	1.21E-03
min	1.45E-200	3.99E-01	6.52E+00	0.00E+00	0.00E+00
max	1.31E-02	4.79E+03	3.61E+01	5.99E-02	2.87E-03

V. CONCLUSIONS

In this paper, we propose two types of variants of PSO: one retains the cognitive constant and further extends the only one social constant to more; the other omits the cognitive constant, but extends the only one social constant to more. For the moment, we only study a class of positive constants c_1 to c_{m+1} for type 1 and c_1 to c_m for type 2, in which a total of 4 was equally distributed to all constants. Better performance for considering some local best particles shows that the two types of variants of PSO have potential advantages over the standard PSO. In the future, we will study how a different proportion of constants affect the performance of optimization and how different functions are improved by these positive constants. Furthermore, introducing more local best particles into other modified versions of PSO is also a big issue.

ACKNOWLEDGMENT

This work is supported by the National Science Council, Republic of China, under Grant NSC 101-2221-E-040-010.

REFERENCES

[1] J. Kennedy and R. Eberhart, “Particle swarm optimization,” Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948, December 1995.

[2] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 69-73, 1998.

[3] Y. Shi and R. Eberhart, “Parameter selection in particle swarm optimization,” Evolutionary Programming VII Lecture Notes in Computer Science, vol. 1447, pp 591-600, 1998.

[4] Y. Shi and R. Eberhart, “Empirical study of particle swarm optimization,” Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1945-1950, 1999.

[5] M. Clerc and J. Kennedy, “The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space,” IEEE Trans. Evol. Comput., vol. 6, pp. 58–73, 2002.

[6] I. C. Trelea, “The particle swarm optimization algorithm: Convergence analysis and parameter selection,” Information Processing Letters 85, pp. 317–325, 2003.

[7] M. Jiang, Y.P. Luo, S.Y. Yang, “Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, ” Information Processing Letters 102, 8-16, 2007.

[8] J. Kennedy, “Bare bones particle swarms,” Proceedings of the IEEE Swarm Intelligence Symposium, pp. 80-87, 2003.

[9] R. A. Krohling and E. Mendel, “Bare bones particle swarm optimization with Gaussian or Cauchy jumps,” Proceedings of the IEEE Congress on Evolutionary Computation, pp. 3285–3291, 2009.

[10] J. Yao and D. Han, “Improved barebones particle swarm optimization with neighborhood search and its application on ship design,” Mathematical Problems in Engineering, vol. 2013, pp. 1-12, 2013.

[11] R. A. Krohling, “Gaussian swarm: a novel particle swarm optimization algorithm,” Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, pp. 372-376, 2004.

[12] R. A. Krohling, “Gaussian particle swarm with jumps” Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1226-1231, 2005.

[13] R. A. Krohling, L. dos S Coelho. “Co-evolutionary particle swarm using Gaussian distribution to solving constraint optimization problems,” IEEE Trans. on Systems, Man, and Cybernetics, part B: Cybernetics, vol. 36, pp.1407-1416, 2006.

[14] N. Higashi, and H. Iba, “Particle swarm optimization with Gaussian mutation,” Proceedings of the IEEE Swarm Intelligence Symposium, pp. 72-79, 2003.

[15] H. Wang, Z. Wu, S. Rahnamayan, C. Li, S. Zeng, and D. Jiang, “Particle swarm optimisation with simple and efficient neighbourhood search strategies,” International Journal of Innovative Computing and Applications, vol. 3, no. 2, pp. 97–104, 2011.

[16] J. kenndy and R. Mendes, “Neighborhood topologies in fully informed and best-of-neighborhood particle swarms,” IEEE Trans. on Systems, Man, and Cybernetics, part C: Applications and Reviews, vol. 36, pp.515-519, 2006.

[17] R. Mendes, J. Kennedy, and J. Neves, “The fully informed particle swarm: simpler, maybe better,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 204–210, 2004.

[18] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization: An overview,” Swarm Intell., vol. 1, pp. 33-57, 2007.