

Verification of a hypothesis about unification and simplification for position updating formulas in particle swarm optimization

Hu Jian

Southwest Nationalities Research Academy, Southwest University for Nationalities, China

hujian@swun.edu.cn

Keywords: particle swarm optimization (PSO); swarm intelligence; artificial intelligence; evolutionary computation; position updating; convergence

Abstract. Particle swarm optimization (PSO) has been a popular research area in artificial intelligence technology, where the two issues of theoretical analysis and premature convergence have been the focus of attention. However, due to complex dynamics in particle swarm, the former has been conducted only in simplified systems. And the latter has been dealt with only by introducing some additional operations, which inevitably increases the complexity of PSO and makes the theoretical analysis more difficult. To handle the above problems, we already proposed a unified and simplified formula for position updating in the existing algorithms, but that formula depends heavily on the hypothesis of that new positions of particles are centered on their weighted experience. In this paper, we selected ten algorithms that were widely used or more novel, and generated a large number of data samples to test their frequency histograms. The experiment results verified this hypothesis, and further proved the correctness of the unification and simplification for position updating formulas.

Introduction

Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart^[1], has been a popular research area in artificial intelligence technology, which has been used to solve a range of optimization problems. In PSO, the two issues of theoretical analysis and premature convergence have been the focus of attention. However, due to complex dynamics in particle swarm, the former has been conducted only in simplified systems, for example, not considering the random number in the position updating formulas^[2], and keeping the previous best position unchanged^[3]. And the latter has been dealt with only by introducing some additional operations, for example, modifying the coefficients' values^[4], changing the communication topologies^[5], and combining genetic operators^[6], which inevitably increases the complexity of PSO and makes the theoretical analysis more difficult.

In order to handle the above problems, reference [7] proposes a unified and simplified formula for position updating in the existing PSO algorithms. That formula simplifies the multiple-order stochastic difference equation to a first-order stochastic difference equation, reduces the multiple parameters to only one parameter, and makes it is easier to analyze the convergence and to control the search behavior of particles. However, in this reference, the process of unification and simplification depends heavily on the hypothesis of that new positions of particles are centered on their weighted experience. In this paper, we will verify this hypothesis, which further proves the correctness of the unification and simplification for position updating formulas.

The rest of this paper is organized as follows. Section 2 describes standard PSO and introduces the unified and simplified position updating formula. This is followed in Section 3 by the methodology to verify the hypothesis. Section 4 is about experimental configuration, results and discussions. Conclusions are given in Section 5.

Standard Particle Swarm Optimization and a Unified and Simplified Position Updating Formula

To build a baseline for performance testing of improvements to PSO, in Proceedings of the 2007 IEEE Swarm Intelligence Symposium, D. Bratton and J. Kennedy^[8] defined the constriction PSO^[9] as the standard PSO (abbreviated as SPSO-SIS). In SPSO-SIS, a particle's position updating formula is often described as the following Eq. 1.

$$X(t) = X(t-1) + \omega * V(t-1) + C_1 * rand_1 * [P_i - X(t-1)] + C_2 * rand_2 * [P_n - X(t-1)] \quad (1)$$

In this notation, V and X are the velocity and position of the current particle; P_i and P_n are the previous best positions of the current particle and of the current particle's neighborhood; ω , C_1 and C_2 are called a inertia weight, a confidence parameter and a social parameter, respectively; $rand_1$ and $rand_2$ are independent uniform random numbers; t is the generation number.

So far, there have been hundreds of modified PSO. Reference [7] unified most of these PSO's position updating formulas into the following formula:

$$X(t) = \sum_j a_j * X(t-j) + \sum_k b_k * P_k, \quad (2)$$

where $X(t)$ and $X(t-j)$ are the current particle's positions in the t -th and the $(t-j)$ -th generation, P_k are is the k -th particle's previous best position, and a_j and b_k are the coefficients. The following equation

$$\sum_j a_j + \sum_k b_k = 1 \quad (3)$$

was proven in reference [7]. Bringing Eq. 3 into Eq. 2, it is gotten that

$$X(t) = \sum_j a_j * X(t-j) + \left(1 - \sum_j a_j\right) * Q = Q + \sum_j a_j * [X(t-j) - Q], \quad (4)$$

wherein,

$$Q = \frac{\sum_k b_k * P_k}{\sum_k b_k}. \quad (5)$$

Based on the hypothesis of that $\sum_j a_j * [X(t-j) - Q]$ is a rand number with a zero mean, reference [7] concluded that the unified and simplified formula for position updating is

$$X(t) = Q + rand(0), \quad (6)$$

where $rand(0)$ denotes a rand number with a zero mean.

Methodology to Verify the Hypothesis

This paper will verify the hypothesis mentioned above, i.e.

$$\sum_j a_j * [X(t-j) - Q] = rand(0). \quad (7)$$

In order to fully verify it, this paper will cover all categories of PSO algorithms, and as same as in reference [7], the following 10 widely used or more novel algorithms are chosen: (1) Standard PSO from IEEE Swarm Intelligence Symposium: SPSO-SIS^[8]; (2) Standard PSO 2007 in Particle Swarm Central: SPSO-2007^[10]; (3) Cooperative PSO: CPSO^[11]; (4) PSO with general fitness evaluation strategy: PSO-GFES^[12]; (5) Fully informed particle swarm: FIPS^[13]; (6) Comprehensive learning PSO: CLPSO^[14]; (7) Orthogonal learning PSO: OLPSO^[15]; (8) Multi-layer particle swarm optimization: MLPSO^[16]; (9) Bare bones PSO: BBPSO^[17]; (10) Quantum-behaved particle swarm optimization: QPSO^[18,19].

In term of the above mentioned algorithms, Eq. 7 can be specified as following, where all symbols denote the same as those in the above section.

In algorithms (1)-(4), the specific hypotheses are

$$\frac{X(t-1) + \omega * V(t-1) + C_1 * rand_1 * [P_i - X(t-1)] + C_2 * rand_2 * [P_n - X(t-1)]}{C_1 * rand_1 * P_i + C_2 * rand_2 * P_n} = rand(0). \quad (8)$$

In algorithm (5), the specific hypothesis is

$$X(t-1) + \chi * \left\{ V(t-1) + \left(\sum_{k=1}^N W_k * \varphi_k * rand_k \right) * [P_m - X(t-1)] \right\} - P_m = rand(0), \quad (9)$$

wherein $P_m = \frac{\sum_{k=1}^N W_k * \varphi_k * rand_k * P_k}{\sum_{k=1}^N W_k * \varphi_k * rand_k}$, N is the number of the current particle's neighbors, and it is met

that $\sum_{k=1}^N W_k * \varphi_k = \varphi$ where W_k is a weighed coefficient.

In algorithms (6) and (7), the specific hypotheses are

$$X(t-1) + \omega * V(t-1) + C * rand * [P - X(t-1)] - P = rand(0), \quad (10)$$

wherein, C is a coefficient. P is determined by means of selecting the best among neighbors in CLPSO, and by means of orthogonal experimental design in OLPSO.

In algorithm (8), the specific hypothesis is

$$X(t-1) + \omega * V(t-1) + \sum_{j=1}^{m-1} \sum_{k=1}^{N_j} \varphi_{j,k} * rand_{j,k} * [P_{j,k} - X(t-1)] - \frac{\sum_{j=1}^{m-1} \sum_{k=1}^{N_j} \varphi_{j,k} * rand_{j,k} * P_{j,k}}{\sum_{j=1}^{m-1} \sum_{k=1}^{N_j} \varphi_{j,k} * rand_{j,k}} = rand(0), \quad (11)$$

wherein m is the number of layers, N_j represents the number of the neighbors at layer j , $P_{j,k}$ is the k -th

neighbor at layer j , and it is met $\varphi_{jk} = \frac{\varphi}{\sum_{j=1}^{m-1} N_j}$.

In algorithm (9), the specific hypothesis is

$$0 = rand(0). \quad (12)$$

It is seen that this hypothesis is always met, by the reason that the zero is a special rand number whose mean and standard deviation are both zeros.

In algorithm (10), the specific hypothesis is

$$\begin{cases} L + \alpha * [M - X(t-1)] * \ln(1/u) - \frac{L + \alpha * \ln(1/u) * M}{1 + \alpha * \ln(1/u)} = rand(0), \\ \quad \text{if } k \geq 0.5, M \geq X(t-1), \text{ or } k < 0.5, M < X(t-1); \\ L - \alpha * [M - X(t-1)] * \ln(1/u) - \frac{L - \alpha * \ln(1/u) * M}{1 - \alpha * \ln(1/u)} = rand(0), \\ \quad \text{if } k \geq 0.5, M < X(t-1), \text{ or } k < 0.5, M \geq X(t-1); \end{cases} \quad (13)$$

wherein $L = \varphi * P_i + (1 - \varphi) * P_g$, g represents the index of the best-performing particle in the whole swarm, P_g is the previous best position obtained by the whole swarm; α is a contraction-expansion coefficient; $M = \frac{1}{N} \sum_{j=1}^N P_j$, P_j denotes the j -th particle's previous best position; φ , k and u are independent uniform random numbers in the range of $[0, 1]$.

To verify the above hypotheses of Eq. 8-11, and Eq. 13, the experiments in the following section will be conducted in corresponding algorithms for generating a large amount of data samples to test their frequency histograms.

Experiments

Test Functions and Algorithms Configuration

Ten benchmark functions are used for experiments because of their popularity in the PSO community. The names and formulas of these functions, the global optimum x^* , the corresponding fitness value $f(x^*)$, and the feasible bounds $[x_{min}, x_{max}]$ are given in Table 1, where D denotes the number of dimensions and is set to 30.

Table 1 Benchmark Functions in the Experiments

Function	x^*	$f(x^*)$	Feasible Bounds	Initialization Range
Sphere (unimodal): $f_1(x) = \sum_{i=1}^D x_i^2$	0^D	0	$[-100,100]^D$	$[50,100]^D$
Rosenbrock's (unimodal): $f_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	1^D	0	$[-30,30]^D$	$[15,30]^D$
Ackley's (unrotated multimodal): $f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	0^D	0	$[-32,32]^D$	$[16,32]^D$
Griewank's (unrotated multimodal): $f_4(x) = \frac{1}{4000} \sum_{i=1}^D \frac{x_i^2}{\sqrt{i}} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0^D	0	$[-600,600]^D$	$[300,600]^D$
Rastrigin's (unrotated multimodal): $f_5(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0^D	0	$[-5.12,5.12]^D$	$[2.56,5.12]^D$
Schwefel's (unrotated multimodal): $f_6(x) = 418.9829 \times D - \sum_{i=1}^D x_i \sin\left(x_i ^{\frac{1}{2}}\right)$	420.96^D	0	$[-500,500]^D$	$[-500,-250]^D$
f_7, f_8, f_9 and f_{10} (rotated multimodal): f_7, f_8 and f_9 are f_3, f_4 and f_5 under coordinate rotation using Salomon's algorithm ^[20] , respectively; f_{10} is f_6 rotated by the methods in reference [14].				

All these algorithms are configured as same as the corresponding references, except that the following configuration need to be clarified:

- (1) The number of particles is set to 50 in FIPS, according to reference [8].
- (2) The velocity is clamped into the range of $[-(x_{max}-x_{min})/2, (x_{max}-x_{min})/2]$ in MLSPSO and QPSO.
- (3) In SPSO-SIS, PSO-GFES, FIPS, CLPSO, OLPSO, MLPSO and QPSO, the position in each dimension is computed iteratively unless the position in the dimension is located inside the feasible search space.
- (4) In OLPSO and QPSO, those parameters decrease with the fitness evaluation number instead of with the generation number, by the reason that the generation number cannot be presetted due to position boundary conditions.
- (5) The best performed algorithms in references, i.e. CPSO-H in reference [11], PSO-P5 in reference [12], FIPS with U-Ring topology in reference [13], OPLSO-L in reference [15], and MLPSO-6L in reference [16], are selected in this paper.
- (6) In QPSO, the value of contraction-expansion coefficient decreases linearly, according to reference [19].
- (7) The termination condition is set to 300,000 fitness evaluations, according to reference [8].

Experimental Results and Discussion

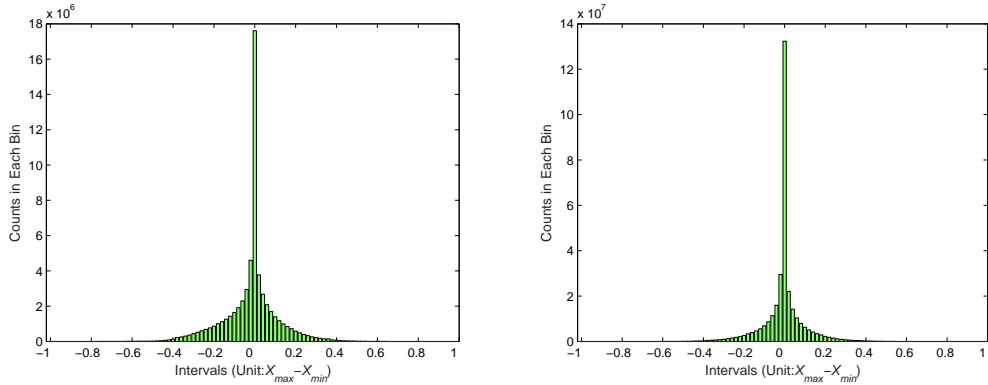
Each algorithm runs 30 times independently. Fig.1(a)-(d) are frequency histograms that count times when each dimension of left side of Eq. 10 is located in per interval of $(x_{max}-x_{min})/101$ after each generation in the first 1,000 fitness evaluations, the first 5,000 fitness evaluations, the first 20,000 fitness evaluations and all 300,000 fitness evaluations, respectively. In order to facilitate the display,

the statistic results in Fig.1 come from all particles in all algorithms and from all benchmark functions.

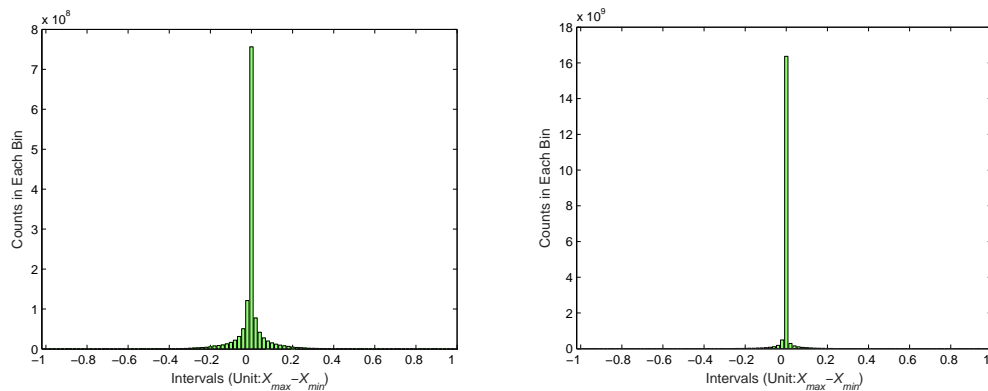
The following phenomena can be seen in Fig.1:

(1) In the whole process of evolution, all samples are centered at zeros and distributed symmetrically. This verifies that $\sum_j a_j * [X(t-j) - Q]$ is a rand number with a zero mean.

(2) From Fig.1 (a) to (d), sample distribution is different, and become more and more gathered toward zeros, with the evolutionary process. This is because that particle swarm tends to explore in the early stage of evolutionary process, and that in the later stage, particle swarm becomes to converge and tends to exploit.



(a) In the first 1,000 fitness evaluations (b) In the first 5,000 fitness evaluations



(c) In the first 20,000 fitness evaluations (d) In the all 300,000 fitness evaluations

Fig. 1. Frequency Histograms of $\sum_j a_j * [X(t-j) - Q]$

Conclusion

In this paper, we represented further investigation on the unification and simplification for position updating formulas^[7] in particle swarm optimization. Aiming at the hypothesis of that new positions of particles are centered on their weighted experience, we selected ten algorithms that are widely used or more novel, and the results of experiments showed the correctness of this hypothesis.

Acknowledgment

This research is supported by Humanities and Social Science Research Youth Fund Project of Ministry of Education in China (Grant No: 15YJC870010), by Scientific Research Fund of Sichuan Provincial Education Department (Grant No: 16ZA0012), and by Graduate Degree Program of Southwest University for Nationalities (Grant No: 2016-XWD-B0304).

References

- [1] J. Kennedy, R. Eberhart. Particle swarm optimization[C]// IEEE International Conference on Neural Networks, 1995. Proceedings. 1995:1942-1948.
- [2] Cleghorn C W, Engelbrecht A P. A generalized theoretical deterministic particle swarm model[J]. *Swarm Intelligence*, 2014, 8(1):35-59.
- [3] Cleghorn C W, Engelbrecht A. Fully informed particle swarm optimizer: Convergence analysis[C]// Evolutionary Computation (CEC), 2015 IEEE Congress on. IEEE, 2015:164-170.
- [4] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[C]// Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. IEEE, 1999:32-49.
- [5] Kennedy J, Mendes R. Population structure and particle swarm performance[C]// Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on. IEEE, 2002:1671-1676.
- [6] Saxena N, Tripathi A, Mishra K K, et al. Dynamic-PSO: An improved particle swarm optimizer[C]//Evolutionary Computation (CEC), 2015 IEEE Congress on. IEEE, 2015: 212-219.
- [7] Hu J, Hu W and Yin F. Unification and Simplification for Position Updating Formulas in Particle Swarm Optimization, unpublished.
- [8] Bratton D and Kennedy J. Defining a standard for particle swarm optimization[C]. Proceedings of the IEEE Swarm Intelligence Symposium. 2007: 120-127.
- [9] Clerc M and Kennedy J. The particle swarm - Explosion, stability, and convergence in a multidimensional complex space[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(1): 58-73.
- [10] PSC, "Particle Swarm Central," 2016, <http://www.particleswarm.info/>. [Online. Last accessed 26-April-2016].
- [11] Van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization[J]. *Evolutionary Computation*, *IEEE Transactions on*, 2004, 8(3): 225-239.
- [12] Hu J, Wang Z, Qiao S, et al. The fitness evaluation strategy in particle swarm optimization[J]. *Applied Mathematics and Computation*, 2011, 217(21): 8655-8670.
- [13] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, maybe better[J]. *Evolutionary Computation*, *IEEE Transactions on*, 2004, 8(3): 204-210.
- [14] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. *Evolutionary Computation*, *IEEE Transactions on*, 2006, 10(3): 281-295.
- [15] Zhan Z H, Zhang J, Li Y, et al. Orthogonal learning particle swarm optimization[J]. *Evolutionary Computation*, *IEEE Transactions on*, 2011, 15(6): 832-847.
- [16] Wang L, Yang B, Chen Y. Improving particle swarm optimization using multi-layer searching strategy[J]. *Information Sciences*, 2014, 274: 70-94.
- [17] Kennedy J. Bare bones particle swarms[C]//Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE. IEEE, 2003: 80-87.
- [18] Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior[C]// Evolutionary Computation, 2004. CEC2004. Congress on. 2004:1571-1580.
- [19] Sun J, Fang W, Palade V, et al. Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point[J]. *Applied Mathematics and Computation*, 2011, 218(7): 3763-3775.
- [20] Salomon R, Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions [J]. *BioSystems*, 1996(39):263-278.