

New Evolution Algorithm Based On The Standard Particle Swarm Optimization

Lipeng Wang¹, Yangjie Cheng², Dong C. Liu³

^{1,2,3} Computer Science College, Sichuan University, Chengdu, China
Email: wlpescu@126.com

Abstract—The particle swarm optimization (PSO) is an alternative for global optimization. A standard for PSO (SPSO) was defined which took into the latest developments, and was used as a baseline for performance testing of improvements. In SPSO, however, particles need to search the optimal solution in the constraint space, and the item velocity in PSO makes particles difficult to adjust themselves to meet those complicated constraints. A new PSO without the item velocity based on SPSO is proposed in this article. The new algorithm inherits the capability of SPSO with fast convergence and high accuracy. This research proves that the convergence process of PSO has nothing to do with the velocity and the proposed method modified simple PSO (msPSO) can converge. The experiments show that msPSO is able to achieve a good result.

Keywords- particle swarm optimization; constraint space; optimal solution

I. INTRODUCTION

Particle swarm optimization (PSO) was first proposed by Eberhart and Kennedy in 1995[1]. For the sake of simplicity, we call the original PSO algorithm traditional particle swarm optimization (TPSO). This method, inspired by simulation of bird flocking or fish schooling, is a stochastic global optimization. In TPSO, each particle adjusts itself in the search space according to the best position it has found so far and the best position of all the particles in the swarm. The iteration will terminate when the particle converges to the global best point. In TPSO, particles represent potential solutions for the optimized problem. In the search process, each particle has its own position and velocity, represented by X and V respectively.

Compared to other evolutionary algorithms, such as genetic algorithms, artificial neural networks, genetic programming and ant colony optimization, PSO has the advantage of easy implementation and fast convergence. It is amenable to solve complex and diverse issues. PSO has been applied in function optimization, artificial neural network, fuzzy system control, machine learning and so on [2].

TPSO is mainly dependent on the experience of the corresponding particle and the particle's companions. Under some extreme conditions, convergence of TPSO will be very slow with low accuracy during the process of iteration; moreover, the particles will always "spiral" around the local extreme value. An inertia coefficient is added to TPSO to increase the probability of jumping out of local minimum [3]. The inertia coefficient can also balance between global and

local search. TPSO is non-linear, and until now there still has not been an effective way to mathematically determine the value of parameters in TPSO to simulate the search process. A fuzzy system was proposed to dynamically adapt the inertia coefficient of TPSO, which could adaptively adjust the coefficients [4]. Another modified PSO (MPSO) was proposed to discard the velocity to get rid of restrictions of the range of velocity and the process was only controlled by the particle's position [5]. PSO has developed a lot with many researchers exploring the concepts and application of the algorithm. Because of different kinds of PSO existing, it is hard for researchers to find a reference to improve. A standard for PSO (SPSO) was defined which took into account current developments. SPSO is regarded as the extension of the TPSO, and can improve performance effectively [6]. The algorithm is intended as a baseline of improvements and provides a valuable point of comparison which can be used for researchers to better test new advances. However, SPSO is still limited to velocity. In some situation, the particle still has the problem with divergence and low accuracy. This research proves that the iteration of SPSO has nothing with the velocity, and proposes a new modified simple PSO (msPSO). The new algorithm removes the constraints of velocity and is proved to converge at a fixed position. The experiment illustrates that msPSO can improve its performance for the five benchmark functions.

The paper is organized as follow: In Section II, the related background is presented in details. In Section III, a new modified particle swarm optimization is proposed. What is more, the research also proves that the convergence process of PSO has nothing to do with the velocity and the proposed method msPSO can converge. Experiments are done on five benchmark functions and results are analyzed in Section IV. Finally we conclude the paper in Section V.

II. RESEARCH BACKGROUND

TPSO is a global optimization algorithm. The search space for a specific problem is represented by a swarm with a great number of particles. During the process of iteration, the position X and velocity V of the corresponding particle are updated according to the particle's individual best position of its experience, denoted by "iBest", and the swarm's best position, represented by "gBest". The criterion of judging the position for a particle is presented by a "fitness" function. TPSO can be formulated as follows:

$$V_{id}(t+1) = \omega V_{id}(t) + c_1 r_1 (P_{iBest}(t) - X_{id}(t)) + c_2 r_2 (P_{gBest}(t) - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

where $i=1,2,\dots,N$, N is the number of particles in the swarm, $d=1,2,\dots,D$, D is the dimension of particles. c_1 and c_2 are nonnegative constants, r_1 and r_2 are random number uniformly distributed in $[0,1]$. $V_{id} \in [-V_{max}, V_{max}]$, when V_{id} is more than V_{max} , set $V_{id} = V_{max}$, and on the other hand, when V_{id} is less than $-V_{max}$, set $V_{id} = -V_{max}$, V_{max} is a pre-set parameter. ω is the inertial weight, which is a constant value distributed in $[0,1]$.

Clerc et.al [7] add a constraint factor α to control V as:

$$X_{id}(t+1) = \alpha X_{id}(t) + V_{id}(t+1) \quad (3)$$

[5] shows the evolution of TPSO has nothing to do with the particle velocity, and proposes a modified PSO (MPSO). In MPSO, the velocity V is removed so the discussion of V is avoided, which can simplify the process of iteration. The algorithm is presented by:

$$X_{id}(t+1) = \omega X_{id}(t) + c_1 r_1 (P_{iBest} - X_{id}(t)) + c_2 r_2 (P_{gBest} - X_{id}(t)) \quad (4)$$

In 2007, Daniel Bratton and James Kennedy designed a standard particle swarm optimization (SPSO), which is the extension of TPSO[6]. SPSO combined the latest developments of PSO, and could improve the performance of iteration effectively. SPSO is denoted by:

$$V_{id}(t+1) = \chi(V_{id}(t) + c_1 r_1 (P_{iBest}(t) - X_{id}(t)) + c_2 r_2 (P_{gBest}(t) - X_{id}(t))) \quad (5)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (6)$$

subject to:

$$\varphi = c_1 + c_2 \quad (7)$$

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (8)$$

The research has found that when $\varphi < 4$, particles will spiral around the best found solution in the search space with no guarantee of convergence. While $\varphi > 4$, SPSO can converged quickly.

SPSO is different from TPSO. The difference is as follows:

1) *Swarm Communication Topology*:

The global topology is used in TPSO, as shown in Fig.1 (a). In this topology, the global extreme is chosen by the comparison of all the particles in the swarm. While in SPSO, a local topology is used, and the global extreme for a specific particle is obtained by comparison of its neighbors, as shown in Fig.1 (b). To distinguish between the two concepts, in this paper, the global extreme obtained by the global topology is called the global optimal solution, while chosen by the local topology is called the local optimal solution.

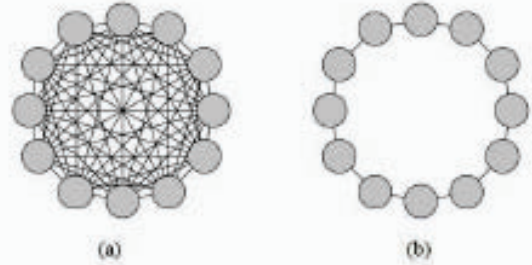


Fig.1. TPSO and SPSO topologies

2) Inertia weight

In TPSO, the inertia weight is used to adjust the influence of velocity in the previous step affected on the present step. By adjusting the inertia factor, particles in TPSO can be bound into the region containing the global optimal solution, while not trapped into local optimal solution. A new variable λ is introduced into SPSO which removes the inertia weight. Experiments show that compared with TPSO, SPSO gets better results while maintains the same simplicity with TPSO. Because SPSO is a combination of the latest PSO developments, it can be used as a reference for future work, thus avoid the redundant work of comparison.

III. METHOD

SPSO includes both velocity (V) and position(X). In practice, due to a lot of restrictions with X , the existence of V makes it very difficult to limit X into a designed search space. In this paper, it is found that evolutionary process of SPSO is independent of V .

Theorem A: SPSO has nothing to do with the particle velocity

Proof: let $\omega = c_1 r_1 \lambda p_0 + c_2 r_2 \lambda p_g$, $\varphi = c_1 r_1 \lambda + c_2 r_2 \lambda$. In SPSO, the constraint factor $\alpha = 1$. [7] has proved that the constraint of V is equivalent to the constraint factor α , so the scope of V will not be considered. Excluding P_{iBest} and P_{gBest} on the impact of each dimension, each dimension is dependent of each other. In order to simplify the certification process, we can just consider the case of one-dimension, and the subscript d can be omitted. Assuming that the other particles are kept fixed except the i -particle, the subscript i can also be omitted. For SPSO, after one iteration, we can get:

$$V(t+2) = \lambda V(t+1) + \omega - \varphi X(t+1) \quad (9)$$

$$X(t+2) = X(t+1) + V(t+2) \quad (10)$$

From equation (9) and (10), we can get:

$$V(t+1) = \frac{1}{\lambda}(X(t+2) - X(t+1) - \omega + \varphi X(t+1)) \quad (11)$$

Put equation(11) into equation(6):

$$X(t+2) + (\varphi - \lambda - 1)X(t+1) + \lambda X(t) - \omega = 0 \quad (12)$$

From equation(12), SPSO has no concept of V, which can simplify the process of PSO. Because SPSO is independent of V, so the item V can be removed from SPSO, just as:

$$X_{id}(t+1) = \chi(X_{id}(t) + c_1 r_1 (P_{iBest} - X_{id}(t)) + c_2 r_2 (P_{gBest} - X_{id}(t))) \quad (13)$$

The research calls this new algorithm modified simple particle swarm optimization (msPSO). In msPSO, the global extreme is obtained through the local topology.

Theorem B: if the average behavior of msPSO is observed through its expected value, msPSO converges

Proof: For ease of exposition, the symbols in theorem A will be used, and the equation(13) can be formatted as:

$$X(t+1) = (\chi - \chi c_1 r_1 - \chi c_2 r_2)X(t) + \chi c_1 r_1 P_{iBest} + \chi c_2 r_2 P_{gBest} \quad (14)$$

The equation (14) is a first order differential equation, and we can set:

$$X(t+1) = X(t) + X'(t)(t+1-t) \quad (15)$$

From the equation(15), the equation (14) can be deformed as:

$$X'(t) + (\chi c_1 r_1 + \chi c_2 r_2 - \chi - 1)X(t) = \chi c_1 r_1 P_{iBest} + \chi c_2 r_2 P_{gBest} \quad (16)$$

Solution for the differential equation is:

$$X(t) = \frac{\chi c_1 r_1 P_{iBest} + \chi c_2 r_2 P_{gBest}}{\chi c_1 r_1 + \chi c_2 r_2 - \chi + 1} + C e^{-(\chi c_1 r_1 + \chi c_2 r_2 - \chi + 1)t} \quad (17)$$

Since the first half items of equation(17) is constant, they can be omitted. If $\lim_{t \rightarrow \infty} X(t)$ converges, then:

$$\lim_{t \rightarrow \infty} \left| \frac{C e^{-(\chi c_1 r_1 + \chi c_2 r_2 - \chi + 1)(t+1)}}{C e^{-(\chi c_1 r_1 + \chi c_2 r_2 - \chi + 1)t}} \right| < 1 \quad (18)$$

Equation (18) is equivalent to equation(19):

$$\chi c_1 r_1 + \chi c_2 r_2 - \chi + 1 > 0 \quad (19)$$

Since r_1 and r_2 are uniformly distributed, the average behavior of msPSO is observed through its expected value.

$$E(r_1) = \int_0^1 \frac{x}{1-x} dx = \frac{1}{2} \quad (20)$$

$$E(r_2) = \int_0^1 \frac{x}{1-x} dx = \frac{1}{2} \quad (21)$$

From the equation(20) and the equation(21), we can infer that the equation(19) is equivalent to equation(22)

$$\chi(c_1 + c_2 - 2) > -2 \quad (22)$$

Since $\chi > 0, c_1 + c_2 > 4$, the equation(22) holds. Equation(13) converges to:

$$\lim_{t \rightarrow \infty} X(t) = \frac{\chi c_1 r_1 P_{iBest} + \chi c_2 r_2 P_{gBest}}{1 - \chi + \chi c_1 r_1 + \chi c_2 r_2} = \frac{2\chi c_1 P_{iBest} + 2\chi c_2 P_{gBest}}{2 - 2\chi + \chi c_1 + \chi c_2} \quad (23)$$

IV. EXPERIMENTS AND RESULTS

To analysis the effect of the proposed algorithm, five benchmark functions are provided for the comparison of particle swarm optimization:

$$\text{Sphere f1: } f_1(x) = \sum_{i=1}^n x_i^2$$

$$\text{Griewank f2: } f_2(x) = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$\text{Rastrigin f3: } f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$$\text{Quadric f4: } f_4(x) = \sum_{d=1}^n \left(\sum_{j=1}^d x_j \right)^2$$

$$\text{Shaffer f5: } f_5(x) = 0.5 + \frac{(\sin \sqrt{x^2 + y^2}) - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$$

The details about the five benchmake functions described as table 1. In table 1, Dim represents the particle dimension, and Range is the search range of particles. Optimal is the extreme value of the corresponding functions and Goal

represents the accuracy of optimization. Four PSO algorithms (MPSO, msPSO, TPSO, SPSO) will be compared using the five functions. To increase the difficulty of the optimization process, the paper expands the dimension of functions and increases the target precision. With these harsh conditions, it can be more easily for us to compare and highlight performance. In our research, the number of particles is 15, and the maximum number of iterations is 500000. After 500000 iterations, iteration will terminate even the result does not meet the target accuracy. For the other parameters in PSO, the recommended parameters in [5] will be used. In MPSO and msPSO, $c_1 = c_2 = 2.05$, $\omega = 1.0$; in TPSO and SPSO, $c_1 = c_2 = 2.05$, $\omega = 0.8$, $V \in [-2, 2]$.

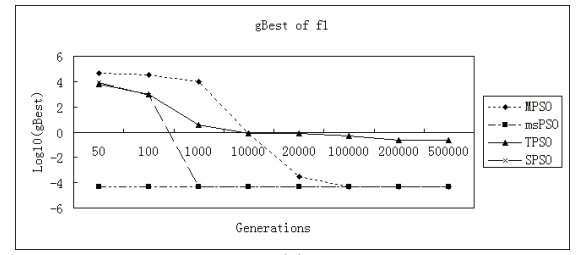
The convergence precision and speed with the fix number of iterations is as shown as fig.2. In Fig.2, X axis represents the iteration number and Y axis represents logarithm of the global optimal solution. When the algorithm reach the required precision after less than 500000 iterations, it will exit and the global optimal.

It can be found in Fig.2 that the five functions can meet the precision requirements with less than 50 iterations. For the function f3, TPSO, SPSO and MSPO always hover around a local minimum, and can not converge; however msPSO can meet the required precision in the number of 50 iterations or less. Compared with other three algorithms, msPSO has a good performance to escape the local minimum. In addition, from the Fig.2, it can be found that for all the objective functions msPSO can achieve the target in less than 500000 iterations, which means that msPSO has good capability of convergence.

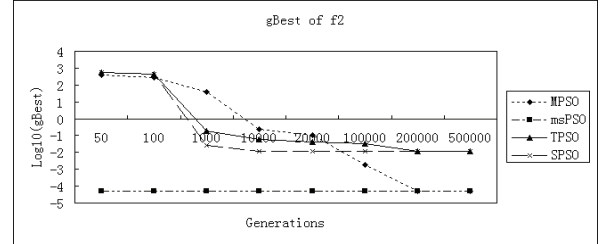
Table 2 shows the number of iterations with the accuracy of 0.00005 for PSO to train the five baseline functions. “>500000” represents that the algorithm is not able to achieve the required precision after 500000 iterations. From Table 2, TPSO, which relates to the selection of too many unknown parameters, is not able to convergence in a fixed number of iterations for f1-f4.

TABLE 1 BENCHMARK FUNCTION

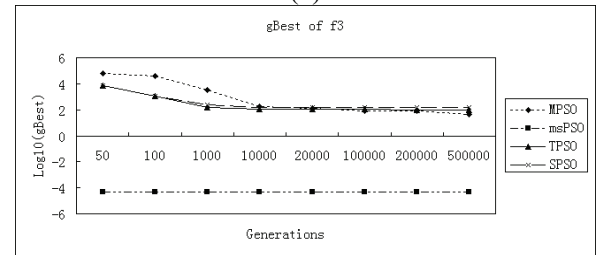
Function name	Dim	Range	Optimal	Goal
Sphere f1	30	$[-100, 100]^{30}$	0	0.00005
Griewank f2	30	$[-600, 600]^{30}$	0	0.00005
Rastrigin f3	30	$[-100, 100]^{30}$	0	0.00005
Duadric f4	30	$[-100, 100]^{30}$	0	0.00005
Shaffer f5	2	$[-100, 100]^2$	0	0.00005



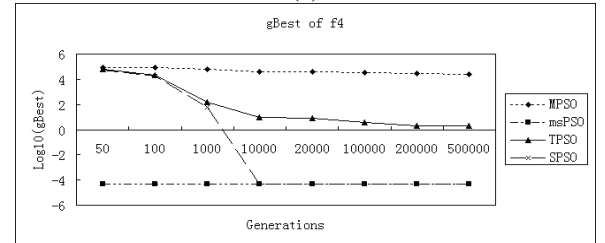
(a)



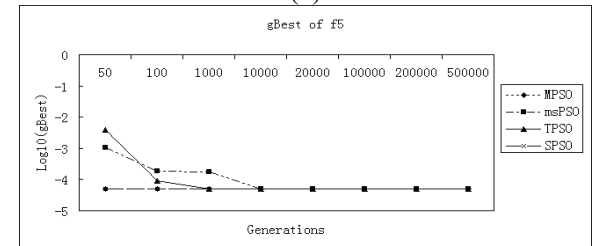
(b)



(c)



(d)



(e)

Fig.2. the evolution curve of the global optimal value for f1-f5

TABLE 2 NUMBERS OF ITERATION UNDER TARGET PRECISION

Function name	Iterations of MSPO	Iterations of msPSO	Iterations of TPSO	Iterations of SPSO
f1	21725	34	>500000	723
f2	123397	35	>500000	>500000
f3	>500000	46	>500000	>500000
f4	>500000	35	>500000	7782
f5	42	3279	105	26

SPSO takes into account the current latest progress of PSO, and its convergence speed has achieved a lot of improvements for f1, f4, f5. Just because of the item V to be considered in SPSO, f2 and f3 still diverge after 500000 iterations. MSPO has been improved a lot to get rid of the constraint of V compared with TPSO, and achieves a good result for f1, f2 and f5. In terms of SPSO msPSO removes the restrictions of V and improve the convergence speed and accuracy.

V. CONCLUSION

In reality, optimization problems face up with many complex and varied constraints. PSO is one of the methods to solve them. However, the velocity V makes X difficult to meet those constraints. The paper proposes msPSO based on SPSO. The new algorithm inherits the capability of SPSO with fast convergence and high accuracy, and gets rid of the item V. This research proves that the convergence process of PSO has nothing to do with V and the proposed method msPSO can converge. The experiments show that msPSO is able to achieve a good result.

REFERENCES

- [1] J. Kennedy and R. Eberhart, "Optimization Particle Swarm, " in Proceedings of the 1995 IEEE International Conference on Evolutionary Computing, Washington DC, 1995, pp. 1942-1948.
- [2] Z. You and T. Ying, "GPU-based Parallel Particle Swarm Optimization," in IEEE Congress on Evolutionary Computation 2009, Trondheim, 2009, pp. 1493-1500.
- [3] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in IEEE Conference on Evolutionary Computation, Anchorage, AK , USA, 1998, pp. 69-73.
- [4] Y. Shi and R. Eberhart, "Fuzzy adaptive particle swarm optimization," in IEEE Conference on Evolutionary Computation, Seoul , South Korea, 2001, pp. 101 - 106.
- [5] H. Wang and L. Zhi-Shu, "A simpler and more effective particle swarm optimization algorithm," Journal of Software, vol. 18, pp. 861-868, 2007.
- [6] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization," in Proceedings of the 2007 Swarm Intelligence Symposium, Honolulu, HI, 2007, pp. 120 - 127.
- [7] M. Clerc, "Towards a deterministic and adaptive particle swarm optimization," in Proceedings of the 1999 Congress on Evolutionary Computation, Washington, 1999, pp. 1951-1957