

One-dimensional Searching-based Particle Swarm Optimization

Wenqiao Lin¹, Yufeng He¹

¹School of Science,
Beijing University of Posts and Telecommunications
Beijing 100876, China
416916073@qq.com

Xinchao Zhao^{1,2}

²State Key Lab. for Novel Software Technology
Nanjing University
Nanjing 210093, China
xcbupt@126.com

Abstract—Particle swarm optimization (PSO) guides its search direction by a linear learning strategy in which each particle updates its velocity through a linear combination among its present status, historical best experience and the swarm best experience. Such a velocity update strategy is easy to achieve, but it is experimentally inefficient when searching in a complex space. The reason is that the current velocity direction of each particle definitely has a great potential on optimal value, however, traditional velocity accumulation search strategy has a great restriction on such a velocity potentiality. Therefore, a new searching mechanism based on One-dimensional Search (OdS) technology is presented in this paper, and a novel PSO variant (OPSO) is also proposed so as to let the swarm effectively search along the first several principal velocity directions by OdS strategy. OPSO can inherit most of the velocity information of all the particles to guide them to the most promising direction, which has a great difference in learning mechanism with usual PSOs. Experimental results indicate that OPSO has competitive performance when comparing with the well-known CMA-ES and CLPSO.

Keywords- Particle swarm optimization; principal component analysis; one-dimensional search; velocity information

I. INTRODUCTION

Nature inspired optimization algorithms, such as genetic algorithm (GA) [1], particle swarm optimization algorithm (PSO) [2], differential evolution (DE) [3], ant colony optimization algorithm (ACO) [4] and artificial bee colony algorithm (ABC) [5], have attracted widely researchers to looking into nature for years—both as model and as metaphor—for inspiration to tackle complex computational problems. A keen observation of the underlying relation between optimization and biological evolution led to the development of an important paradigm of computational intelligence—the evolutionary computing techniques for performing very complex search and optimization.

As a versatile and efficient new technique based on swarm intelligence algorithm for global numerical optimization, PSO has attracted increasing attentions and has been widely used in different application fields. The outstanding feature of PSO is its learning mechanism which distinguishes it from other biological-inspired optimization techniques. In 1995, Kennedy and Eberhart [2] put forward this learning strategy by using a linear combination of these three terms and storing the learning experience in current position. Many improved learning strategies have been proposed since its emergence. Liang et al. [6] proposed a

comprehensive learning strategy aiming to offer a better performance for multimodal functions. Wang et al. [7] employed a generalized opposition-based learning (GOBL) and Cauchy mutation to provide a faster convergence and help particles escape from local optima. Cho et al. [8] presented another enhanced particle swarm optimization using some deterministic samplings to generate new particles for finding multiple local optima in objective function surfaces. Nickabadi et al. [9] proposed a new adaptive inertia weight by using the success rate of the swarm as its feedback parameter to ascertain the particles' situation in the search space. Zhan et al. [10] applied an orthogonal learning strategy to improve the performance of PSO algorithm (OLPSO). And Zhang et al. [15] put forward one of the most salient and active DE research topics by designing a new learning strategy that can utilize previous search information more efficiently (JADE).

It is well known that there are great potentials for particles to find optimal solution along the traditional velocity direction. However, all the learning experience collected from various learning strategies of all the PSO variants are scarcely aimed at enhancing such a learning potential until now. The reason may be that it will take plenty of time complexity for each particle to explore a better solution along its velocity direction.

To overcome this difficulty, a statistical method of principal component analysis (PCA) [11], which can reduce the dimension of the handled multivariate data to some extent, is originally introduced to particle swarm optimization algorithm (denoted as OPSO) trying to exert such a velocity potential abundantly in this paper. PCA technique [11] can decompose the original sample data into all the principal component directions, which are orthogonal to each other. So, what we need to do is to guide some particles to explore along the first several principal directions of velocity information. This exploring operation is accomplished by the well-know One-dimensional Search (OdS) strategy and global best solution is chosen to explore.

The rest of the paper is organized as follows. The framework of standard PSO, the statistical method of principal component analysis and the one-dimensional search strategy are introduced in Section 2. How PSO is combined with PCA and OdS strategy is proposed and analyzed in Section 3. In Section 4, experimental comparisons are conducted to verify the efficacy and efficiency of OPSO. Finally, conclusions and possible future research are given in Section 5.

II. BACKGROUND AND RELATED WORKS

A. Problem Description

Without loss of generality, in this work, we consider the following numerical optimization problem:

Minimize:

$$y=f(x), \quad x \in S \quad (1)$$

where $S \in R^D$ is a compact set, $x=(x_1, x_2, \dots, x_D)^T$ is decision variable, and D is the dimensionality, i.e., the number of decision variables. Generally, for each variable x_j , it satisfies a boundary constraint, such that:

$$L_j \leq x_j \leq U_j, \quad j=1,2,\dots,D \quad (2)$$

B. Standard Particle Swarm Optimization

A standard PSO [2] is an optimization technique based on the cooperation and competition among individuals to complete searching the optimal solution in a d -dimensional hyperspace. There is a swarm of particles and each individual has a fitness value which is decided by the objective function. During the particles evolution, each particle has a velocity vector $V_i=(v_{i1}, v_{i2}, \dots, v_{id})^T$ and a position vector $X_i=(x_{i1}, x_{i2}, \dots, x_{id})^T$ fly to the potential optimal position under the guidance and heuristic information, where i is a positive integer indexing the particle in the swarm and d is the dimension size of the searching hyperspace. Moreover, particle tracks two extremes to update itself. One is its personal historical best position vector P_i and the other is the best position found by the entire swarm, which is denoted as P_g . The vector v_i and the position x_i are randomly initialized and updated by the follow formulas through the guidance of P_i and P_g

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (P_{id}(t) - x_{id}(t)) + c_2 r_2 (P_{gd}(t) - x_{id}(t)) \quad (3)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \quad (4)$$

where ω is the inertia weight, coefficients c_1 and c_2 are the cognitive and social weights, and then r_1, r_2 are two uniform random numbers within the range of $[0, 1]$.

C. One-dimensional Search Strategy

For the real parameter optimization (1), during the iteration k , a straight line that cross point $x^{(k)}$ and down the direction $d^{(k)}$ is denoted as

$$L = \{x | x = x^{(k)} + \lambda d^{(k)}, -\infty < \lambda < +\infty\} \quad (5)$$

Then, to solve the minimum value of problem $f(x)$ along the direction of L is transformed into the following unary function minimizing problem,

$$\varphi(\lambda_k) = \min_{\lambda \in L} f(x^{(k)} + \lambda d^{(k)}) \quad (6)$$

If λ_k is the minimum value of above problem, then the minimum value of original problem $f(x)$ at iteration k along direction L can be denoted as

$$x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)} \quad (7)$$

D. Principal Component Analysis Strategy

Now we consider the s -variable linear transformations [11] as follows:

$$\begin{cases} Z_1 = a'_1 X = a_{11} X_1 + a_{21} X_2 + \dots + a_{s1} X_s, \\ Z_2 = a'_2 X = a_{12} X_1 + a_{22} X_2 + \dots + a_{s2} X_s, \\ \dots\dots\dots \\ Z_s = a'_s X = a_{1s} X_1 + a_{2s} X_2 + \dots + a_{ss} X_s, \end{cases} \quad (8)$$

where a_i is the coefficient of the linear transformation. Here, the covariance matrix Σ of the s variables is used.

Then, the above linear transformation model can be simplified as follows:

$$\begin{aligned} Var(Z_i) &= a'_i \Sigma a_i, & i &= 1, 2 \dots s \\ Cov(Z_i, Z_j) &= a'_i \Sigma a_j, & i, j &= 1, 2 \dots s \end{aligned} \quad (9)$$

The goal of PCA is to transform a set of correlated entire exemplar x_i into several minimally correlated exemplars Z_i ($i=1,2,\dots,s$). To characterize the amount of information that the principal component carries, its variance $Var(Z_i)$ is selected. Probability theory [12] tells us that $Var(Z_i)$ will approach to infinity if no restrictions on the coefficient a_i of the linear combination. This is accomplished by using an orthogonal restriction on a_i , namely, $a'_i a_i = 1$ ($i=1,2,\dots,s$).

When the first principal component reflects the entire exemplar infrequently, another principal component Z_j is used. However, Z_i and Z_j ($i \neq j$) may contain the same information about the entire exemplar, which is not what we want. Therefore, an additional constraint is used as $Cov(Z_i, Z_j) = a'_i \Sigma a_j = 0$, which makes the reflected information in the i -th principal component not appearing in the j -th one and vice versa. Then, the problem of solving the i -th principal component is transformed into a mathematical model as follow:

$$\begin{aligned} &\max Var(a'_i X) \\ &s.t. \begin{cases} a'_i a_i = 1 \\ a'_i \Sigma a_j = 0 \end{cases} \end{aligned} \quad (10)$$

This is an extreme problem with two constraints, which can be solved by Lagrange multiplier method. To solve the first principal component Z_1 , the first constraint is effective. Then we have

$$\varphi(a_1) = Var(a'_1 X) - \lambda(a'_1 a_1 - 1) = a'_1 \Sigma a_1 - \lambda(a'_1 a_1 - 1) \quad (11)$$

It is obvious that $a_i \neq \bar{0}$, then we have $|\Sigma - \lambda I| = 0$. That is to say, in order to solve the first principal component Z_1 , what we need to do is to solve the eigenvector of the covariance matrix of X . Then the objective function is changed into:

$$Var(a'_i X) = a'_i \Sigma a_i = a'_i \lambda_i a_i = \lambda_i \quad (13)$$

which means that maximizing $Var(a'_i X)$ is equivalent to solve the eigenvector of the largest eigenvalue. The other principal component Z_j can similarly be obtained by solving the eigenvector corresponding to the j -th largest

eigenvalue. Furthermore, the second constraint condition in modal (10) will be satisfied automatically if both a_i and a_j are the different eigenvectors of covariance matrix of Σ .

III. IMPROVED PSO WITH PCA AND ODS

In this paper we will combine the principal component analysis technique and One-dimensional search strategy with particle swarm optimization (PSO). First, we describe the motivations of this paper. Then, how PCA and Ods strategy are cooperatively combined with PSO is illustrated.

A. Motivation

PSO relies on its learning strategy to guide its search direction. Traditionally, each particle utilizes its historical best experience and the swarm best experience through linear summation, and flies to the region in a random level based on such a linear summation. Obviously, the traditional velocity direction has an enormous potential for searching the optimal solution. Hence, designing learning strategies that can utilize this potential search direction more efficiently and abundantly has become one of the most salient and active research topics [10].

To abundantly utilize the potential searching ability of the traditional velocity directions, a novel idea comes into mind that particles should have potential capabilities search everywhere along the promising searching directions found by principal component analysis technique.

B. How they cooperate and benefit each other

Obviously, above mind is a greedy and inefficiency search strategy if each particle search everywhere along the velocity direction. However, the appearance of PCA technology makes it possible in this paper.

Multivariate statistical analysis indicate that PCA technology can linearly decompose the original multivariate data into its all principal component directions orthogonal to each other and keep the first several principal directions own almost all the information of the original data. Here, a sample about PCA technology is given to reveal its magical charm for PSO to search the optimal solution.

Suppose that the originate points of each particle's velocity are fixed on the coordinate origin, then we can gain its two principal components Z_1, Z_2 in the following graphics for a 2-dimension realistic problem in one iteration.

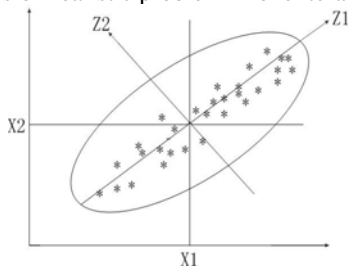


Figure 1. Example of PCA in 2-dimension space.

The above graphics tell us the flying direction of the whole particles can be recapitulated into the first principal direction Z_1 . That means that it only needs to guide one particle to explore along the first principal direction, instead of each particle exploring along its velocity direction. In this paper, we choose the global best particle to explore a better direction along the first several principal directions. Besides the present analysis, numerical experiments also illustrates that such an exploration is effective and feasible.

C. Procedure of OPSO and Pseudo code

- Initialize the position, velocity and parameters;
- Initialize personal historical best value;
- Define each particle's best neighbor by a ring topology and goes into the following loop:
- LOOP: Update particles' positions and velocities based on equations (3) and (4);
- Calculate the first several principal components of the swarm velocity based on cumulative contribution rate;
- Search along the just found principal directions with one-dimensional searching strategy;
- If even better global optimal solution is found, then preserve and update it;
- Update personal historical and neighbor best positions;
- Check the position and velocity boundaries
- If the termination criterion is satisfied, output the best solution. Otherwise, go to LOOP.

IV. EXPERIMENTAL COMPARISON

Besides the above theoretic analysis on OPSO, simulation experiments are presented and compared with other two well-known algorithms in this section to verify the performance of OPSO based on a set of 6 benchmark functions provided by CEC2005 on real parameter optimization [13]. Parameters are set to $\omega=0.725$, and $c_1=c_2=2.0$.

One comprehensive learning particle swarm optimization variant (CLPSO) [14] and a well-known evolution strategy algorithm (CMA-ES) [16] are used for comparison in this section. PSO has been arousing great research interests and has also been successfully applied in various areas since its emergence. CLPSO [14] enhances the population diversity by a novel learning strategy whereby all other particles' historical best information is used to update a particle's velocity, and it is also the prime compare objects of some excellent algorithms. The CMA-ES, which is the best optimization algorithm in CEC'05 special session on real parameter optimization, is also compared here to demonstrate the excellent performance of OPSO algorithm on real parameter optimization. They are the most classical and efficient population-based optimization algorithms. Experimental comparison is show in Table I and Figure 2.

TABLE I. NUMERICAL RESULTS

	CLPSO	CMA-ES	OPSO
	Mean Error \pm Std Dev	Mean Error \pm Std Dev	Mean Error \pm Std Dev
F1	0 \pm 0	5.79e-026 \pm 6.37e-027	2.19e-027 \pm 8.30e-027

F2	856.04±188.32	1.60e-025±2.27e-026	4.32e-023±7.41e-023
F3	1.57e+07±4.11e+06	1.55e-021±1.99e-022	1.98e+06±4.20e+06
F4	6918.87±1821.008	6.90e+04±1.19e+05	132.377±448.8841
F5	3795.455±468.6579	1.31e-010±1.29e-011	7547.2044±2276.4874
F6	5.1425±5.2475	0.478±1.3222	17.5615±4.7577

For a fair comparison among OPSO, CLPSO and CMA-ES, the corresponding parameters are set as the corresponding references. The final solutions are collected over 25 independent runs and summarized in Table I and Figure 2, where “Mean Error”, “Std Dev” mean the average error comparing with real global best solution and the standard deviation of the 25 final results in 25 runs respectively. Observed from above table, it can be seen that OPSO competitively and complementarily performs as well as CMA-ES in terms of the solution accuracy and stability.

V. CONCLUSION

An improved PSO with PCA and OdS strategy is presented in this paper aiming at further refining the final solution of classic PSO algorithm and enhancing its adaptability in higher dimensions and hyper searching space. In this algorithm, the first six benchmark functions provided by CEC’05 and two well-known algorithms are used to compare the performance. Experimental results illustrate its great capacity and reliability on real optimization problem.

Future research will be undertaken to elaborate the validity of the OdS strategy based on PCA technology, particularly when searching in a higher hyperspace.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China (61105127).

REFERENCES

[1] J. Holland, “Adaptation in Natural and Artificial Systems”, MIT Press, Cambridge, MA, 1992.

[2] J. Kennedy and R. C. Eberhart, “Particle swarm optimization”, in Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, 1942-1948, 1995.

[3] S. Das, P. N. Suganthan, “Differential Evolution: A survey of the State-of-the-Art”. IEEE Trans. Evol. Comput., 15(1):4-31, 2011.

[4] P. Li, K. Song, E. Yang, “Quantum ant colony optimization with application,” International Conference on Natural Computation (ICNC), vol. 6, pp.2989-2993, 2010.

[5] X. Y.Li, Z. C. Li and L. Q. Lin, “An artificial bee colony algorithm for multiobjective optimization”. International Conference on Intelligent Systems Design and Engineering Application,153-156, 2012.

[6] J.J. Liang, A.K. Qin, S. Baskar, “Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions”, IEEE Trans. Evol. Compu. 10(3): 281-295, 2006.

[7] H. Wang, Z. J. Wu, Y. Liu, “Enhancing particle swarm optimization using generalized opposition-based learning,” Inform. Sci. 181(2011) 4699-4714.

[8] H. Cho, D. Kim, F. Olivera, S. D. Guikema, “Enhanced speciation in particle swarm optimization for multi-modal problems”. European Journal of Operational Research. 213(2011) 15-23.

[9] A. Nickabadi, M. M. Ebadzadeh, R. Safabakhsh, “A novel paticle swarm optimization algorithm with adaptive inertia weight”, Applied Soft Computing. 11(2011) 3658-3670.

[10] Z. H. Zhan, J. Zhang, Y. Li, Y. H. Shi, “Orthogonal learning particle swarm optimization”, IEEE Trans. Evolut. Comput. 15(2011) 832-847.

[11] H. X. Gao, Applied Multivariate Statistical Analysis, Peking University Press, 2005.

[12] Y. X. Wang, H.X. Sun, Probability theory and stochastic processes, Beijing University of Posts and Telecommunications Press, 2003.

[13] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Nanyang Technol. Univ., Singapore, 2005.

[14] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” IEEE Trans. Evolut. Comput, vol. 10, no. 3, pp. 281-295, 2006.

[15] J. Zhang and A. C. Sanderson, “JADE: Adaptive differential evolution with optional external archive,” IEEE Trans. Evol. Comput, vol. 13, no. 5, pp. 945-958, 2009.

[16] A. Auger and N. Hansen, “A restart CMA evolution strategy with increasing population size,” Proc. IEEE CEC, 2005, 1769-1776.

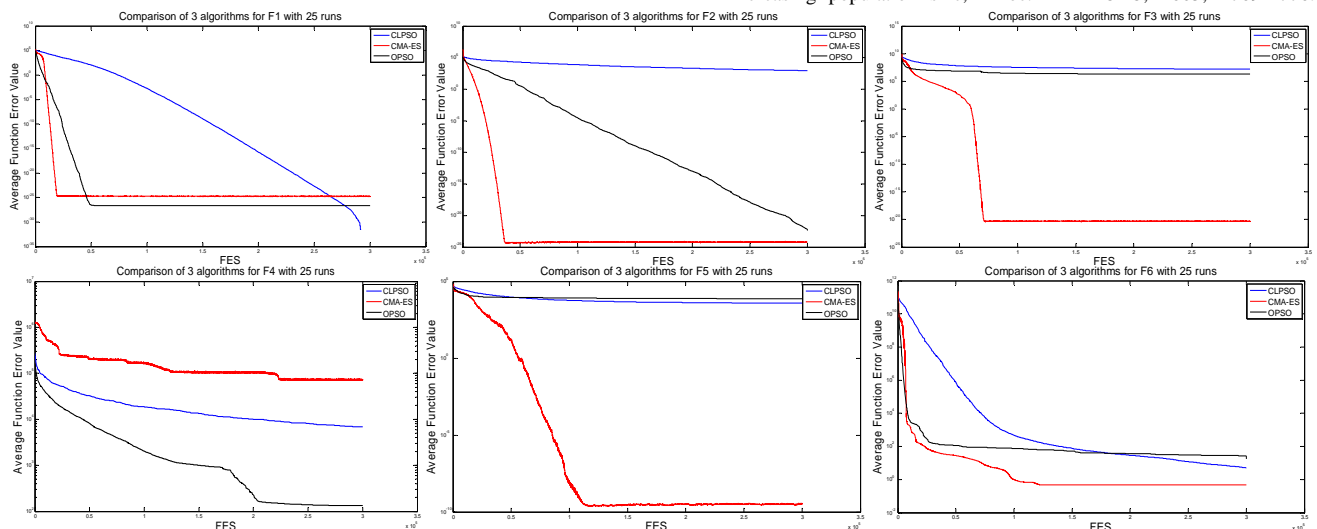


Figure 2. The evolution procedure of six CEC 2005 benchmark functions