Available online at www.sciencedirect.com
ScienceDirect
Information Processing Letters 105 (2008) 231–235
Information Processing Letters
www.elsevier.com/locate/ipl

Dispersed particle swarm optimization

Xingjuan Cai [a,*], Zhihua Cui [b,a], Jianchao Zeng [a], Ying Tan [a]

[a] *Division of System Simulation and Computer Application, Taiyuan University of Science and Technology, Taiyuan 030024, P.R. China*
[b] *State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710049, P.R. China*

Received 20 March 2007; received in revised form 14 August 2007; accepted 5 September 2007

Available online 11 September 2007

Communicated by F. Meyer auf der Heide

Abstract

In particle swarm optimization (PSO) literatures, the published social coefficient settings are all centralized control manner aiming to increase the search density around the swarm memory. However, few concerns the useful information inside the particles' memories. Thus, to improve the convergence speed, we propose a new setting about social coefficient by introducing an explicit selection pressure, in which each particle decides its search direction toward the personal memory or swarm memory. Due to different adaptation, this setting adopts a dispersed manner associated with its adaptive ability. Furthermore, a mutation strategy is designed to avoid premature convergence. Simulation results show the proposed strategy is effective and efficient.

Keywords: Particle swarm optimization; Social coefficient setting; Dispersed control; Centralized control; Adaptation

1. Introduction

Particle swarm optimization (PSO) [1,2] is a new swarm intelligent technique inspired by the birds flocking and fish schooling. Due to the fast convergence speed and easy implementation, it has gained much attention and wide applications in many areas such as neural network, dynamic web organizing, fitness prediction, mountain clustering, parameter selection, etc. [3–8].

As a population-based evolutionary technique, each individual (called particle) searches the multi-dimensional domain space with position and velocity information, and preserve the best position found by itself. Suppose $v_{jk}(t)$ is the kth dimensional value of velocity vector of particle j at time t, then $v_{jk}(t+1)$ is calculated as follows:

$$v_{jk}(t+1) = wv_{jk}(t) + c_1 r_1\big(p_{jk} - x_{jk}(t)\big) + c_2 r_2\big(p_{gk} - x_{jk}(t)\big) \quad (1)$$

where p_{jk} and p_{gk} are the kth dimensional values of historical best positions about particle j and the swarm, in other words, they are kth dimensional values of particle j's memory and the swarm memory. $x_{jk}(t)$ is the kth value of position of particle j. r_1 and r_2 are two random numbers generated with uniform distribution within 0 and 1.

To control excessive roaming of particles outside the search space, $v_{jk}(t+1)$ is limited by:

$$\big|v_{jk}(t+1)\big| \leqslant v_{\max} \quad (2)$$

* Corresponding author.
E-mail addresses: cai_xing_juan@sohu.com (X. Cai), cuizhihua@gmail.com (Z. Cui), zengjianchao@263.net (J. Zeng).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2007.09.001

where velocity threshold $v_{\max}$ is a predefined constant. Then, the position value $x_{jk}(t+1)$ is updated with

$$x_{jk}(t+1) = x_{jk}(t) + v_{jk}(t+1) \tag{3}$$

PSO consists of three parameters: inertia weight, cognitive and social coefficients, and many experiments have demonstrated the social coefficient plays a more important role in the search efficiency. In 1995, J. Kennedy [1] pointed that the coefficients of cognitive and social should be the same, and suggested both setting to 2.0, although E. Ozcanand [9] proposed one different setting $c_1 = c_2 = 1.494$. To further enhance the search density surrounding the historical best position found by the swarm, A. Ratnaweera [10] published one dynamic strategy that cognitive coefficient decreased linearly from 2.5 to 0.5, while social coefficient increased from 0.5 to 2.5. Similarly, G. Venter [11] also noted that small cognitive coefficient and large social coefficient can improve the performance greatly. With statistic analysis, Y. Peng [12] proved that the performance of PSO was mainly affected by social coefficient.

All of these settings are all particle-independent, in other words, centralized control—the same value among the swarm in each generation. In this manner, the swarm tends to search around the historical best position of the swarm $\overrightarrow{p_g}$. Then, some useful information inside the personal historical best position $\overrightarrow{p_j}$ may lose, then decrease the search efficiency. To overcome this shortcoming, a dispersed control manner is introduced, in which each particle selects its social coefficient value to decide the search direction: $\overrightarrow{p_j}$ or $\overrightarrow{p_g}$.

The rest of this paper is organized as follows. Section 2 analyzes the shortcomings of the centralized setting about social coefficient, then the details of dispersed particle swarm optimization (DPSO) are explained. Finally, several benchmark functions prove the proposed method is effective and efficient.

2. Disadvantages of centralized social coefficient setting

Generally, the centralized social coefficient setting makes the particles converge onto one position $\overrightarrow{p_g}$. Thus, some useful information among $\overrightarrow{p_j}$ are neglected. The following part provides a detailed discuss about this disadvantages.

2.1. Biology analysis

Partly due to the differences among individuals, swarm collective behaviors are complex processes. For a group of birds or fish families, there exist many differences. Firstly, in nature, there are many internal differences among birds (or fish), such as ages, catching skills, flying experiences, and muscles' stretching, etc. Furthermore, the lying positions also provide an important influence on individuals. For example, individuals, lying in the side of the swarm, can make several choices differing from center others. Both of these differences mentioned above provide a marked contribution to the swarm complex behaviors.

For standard particle swarm optimization, each particle maintains the same flying (or swimming) rules according to (1), (2) and (3). At each iteration, the social coefficient c_2, an important parameter affecting the performance, is the same within the whole swarm of standard PSO, thus the differences among particles are omitted. Since the complex swarm behaviors can emerge the adaptation, a more precise model, incorporated with the differences, can provide a deeper insight of swarm intelligence, and the corresponding algorithm may be more effective and efficient. Inspired with this method, we propose a new dispersed social coefficient setting.

2.2. Trajectory analysis

For particle j, the trajectory of the position vector is [13]

$$\lim_{t\to+\infty} \overrightarrow{x_j}(t) = \frac{c_1 r_1 \overrightarrow{p_j} + c_2 r_2 \overrightarrow{p_g}}{c_1 r_1 + c_2 r_2} \tag{4}$$

where $\overrightarrow{p_j}$, and $\overrightarrow{p_g}$ are temporarily setting as constants independent with time t. Since r_1 and r_2 are random numbers uniformly distributed within $(0, 1)$, the expected value of $\overrightarrow{x_j}(t)$ converges onto

$$\lim_{t\to+\infty} E\left(\overrightarrow{x_j}(t)\right) = \frac{c_1 \overrightarrow{p_j} + c_2 \overrightarrow{p_g}}{c_1 + c_2} \tag{5}$$

Eq. (5) implies the trajectory of particle j converges onto a weighted mean of personal historical best position $\overrightarrow{p_j}$ and the best position $\overrightarrow{p_g}$ found by swarm, as well as the weights of $\overrightarrow{p_j}$ and $\overrightarrow{p_g}$ are $\frac{c_1}{c_1+c_2}$, and $\frac{c_2}{c_1+c_2}$, respectively.

Now, let us consider the action of social coefficient. If $c_2 \to 0$, then

$$\lim_{t\to+\infty} E\left(\overrightarrow{x_j}(t)\right) \to \overrightarrow{p_j} \tag{6}$$

On the contrary, if $c_2 >> c_1$, then

$$\lim_{t\to+\infty} E\left(\overrightarrow{x_j}(t)\right) \to \overrightarrow{p_g} \tag{7}$$

This is the base of the methods proposed by A. Ratnaweera [10] and G. Venter [11], and the algorithm performance is improved greatly. Formulas (6) and (7)

illustrate social coefficient c_2 can determine the search direction. For each particle, due to the biological background mentioned above, each particle should select its social coefficient associated with the performance. Following this idea, we proposed a probability selection mechanism in which each particle can select its convergent objective by different probabilities.

3. Dispersed particle swarm optimization

Without loss of generality, this paper considers the following problem:

$$\min f(X) \quad X \in D \subseteq R^n \tag{8}$$

As a new modified version of PSO, dispersed particle swarm optimization simulates the natural swarm behaviors, and should consider two problems listed as follows:

(1) Since the social coefficient setting of each particle is associated with the performance itself. How to choose the performance and design an index to measure this performance?

(2) How to design the social coefficient setting strategy based on the proposed index?

3.1. Definition of index to measure the performance

Because the fitness value of current position of each particle represents its adaptive capability, it is naturally selected as the performance to distinguish the characteristic differences. Thus, the better the fitness of current position is, the more adaptation that particle can get to enhance its survival ratio.

Since the literatures only consider the extreme value $\overrightarrow{p_g}$, however, they neglect the differences between $\overrightarrow{p_j}$ and $\overrightarrow{p_g}$. These settings lose some information maybe useful to find the global optima or escape from a local optima. Thus, we design a new index by introducing the performance differences, and the definition is provided as follows:

$$\text{Grade}_u(t) = \frac{f_{\text{worst}}(t) - f(x_u(t))}{f_{\text{worst}}(t) - f_{\text{best}}(t)} \tag{9}$$

where $f_{\text{worst}}(t)$ and $f_{\text{best}}(t)$ are the worst and best fitness values of the swarm at time t, respectively. Occasionally, the swarm converges onto one point, that means $f_{\text{worst}}(t) = f_{\text{best}}(t)$. In this case, the value $\text{Grade}_u(t)$ of arbitrary particle u is set to 1. $\text{Grade}_u(t)$ is an information index to represent the differences of particle u at time t, according to its fitness value of the current position. The better the particle is, the larger $\text{Grade}_u(t)$ is, and vice versa.

3.2. Dispersed social coefficient setting strategy

As we known, in most cases, if the fitness value of particle u is better than which of particle m, the probability that global optima falls into u's neighborhood is larger than that of particle m. In this manner, the particle u should pay more attentions to exploit its neighborhood. On the contrary, it may tend to explore other region with a larger probability than exploitation. Thus, for the best solution, it should make complete local search around its historical best position, as well as for the worst solution, it should make global search around $\overrightarrow{p_g}$. Then, the dispersed social coefficient of particle j at time t is set as follows:

$$c_{2,j}(t) = c_{\text{low}} + (c_{\text{up}} - c_{\text{low}})\text{Grade}_j(t) \tag{10}$$

where c_{up} and c_{low} are two predefined numbers, and $c_{2,j}(t)$ represents the social coefficient of particle j at time t.

3.3. Mutation strategy

Since an explicit selection pressure $\text{Grade}_j(t)$ is introduced, the new social coefficient setting encounters local optima with a more probability than the standard version of PSO. To avoid premature convergence, a mutation strategy is introduced to enhance the ability escaping from the local optima.

This mutation strategy is designed as follows. At each time, particle j is uniformly random selected within the whole swarm, as well as the dimensionality k is also uniformly random selected, then, the $v_{jk}(t)$ is changed as follows.

$$v_{jk}(t) = \begin{cases} 0.5 \times x_{\max} \times r_1, & \text{if } r_2 < 0.5, \\ -0.5 \times x_{\max} \times r_1, & \text{otherwise} \end{cases} \tag{11}$$

where r_1 and r_2 are two random numbers generated with uniform distribution within 0 and 1. In each generation, only one dimensional value of one particle is applied with mutation strategy, thus, the mutation probability is $\frac{1}{\text{Popsize}\cdot\text{Dimension}}$.

3.4. The details of the steps of DPSO

Step 1. Initializing the position and velocity vectors of the swarm, and determining the historical best position $\overrightarrow{p_g}$ and $\overrightarrow{p_j}$ $(j = 1, 2, \ldots, n)$;

Step 2. Calculating the dispersed social coefficient according to formula (9) and (10);

Step 3. Updating the position and velocity vectors with formula (1), (2) and (3), whereas the social coefficient c_2 is replaced by $c_{2,j}(t)$;

Step 4. Updating the historical best position $\overrightarrow{p_g}$ and $\overrightarrow{p_j}$ $(j = 1, 2, \ldots, n)$;

Step 5. Making mutation strategy;

Step 6. If the stop criteria is satisfied, output the fitness value of $\overrightarrow{p_g}$; otherwise, goto Step 2.

4. Simulation results

In order to certify the effectiveness of dispersed particle swarm optimization (DPSO), we select four famous benchmark functions: Schwefel Problem 2.26, Rastrigin, Ackley, and Penalized Function to test the performance. The comparison is taken among DPSO and standard PSO (SPSO) and modified PSO with time-varying accelerator coefficients (MPSO-TVAC) [10]. The details of these test functions can be found in [14].

The coefficients are set as follows: the inertia weight w is decreased linearly from 0.9 to 0.4. In MPSO-TVAC, c_1 is decreased from 2.5 to 0.5,while c_2 increased from 0.5 to 2.5. For DPSO, from a large amount of experiments, c_1 is set to 2.0, as well as c_{up} and c_{low} are set as 2.0 and 1.0, can make a better algorithm performance. Total individual is 100, and the dimensionality is 30, 50, 100, 200 and 300, respectively. The velocity threshold $v_{\max}$ is set to the upper bound of domain. In each experiment,the simulation runs 20 times, while each time the largest evolutionary generations are $50 \times$ dimension. For convenience, the position vector of each particle is randomly selected within the search domain, while the velocity vectors are chosen within the interval $[0, v_{\max}]$ randomly.

Schwefel Problem 2.26 is an interesting function, while the global optima lies in $(420.9687, 420.9687, \ldots, 420.9687)$. From Table 1, the performance of DPSO always surpasses than MPSO-TVAC and SPSO for nearly 25%, no matter from mean value and standard deviation value. For Rastrigin Function, DPSO is better than MPSO-TVAC and SPSO from 25–50%. However, Schwefel Problem 2.26 and Rastrigin have little linkages among different dimensional values. It means the proposed DPSO can improve the algorithm performance than MPSO-TVAC at least above 25%.

Ackley and Penalized are two famous benchmark functions with a strong linkages among dimensional values. From Tables 3 and 4, the algorithm performance of DPSO is more stable than MPSO-TVAC and SPSO. It is interesting that the performance of DPSO changes within the same degree when dimensionality is no less than 100.

Table 1
The comparison results of Schwefel Problem 2.26

Dim.	Alg.	Mean	Std.
30	SPSO	−6.72e+003	1.02e+003
	TVAC	−6.62e+003	6.15e+002
	DPSO	−8.58e+003	4.63e+002
50	SPSO	−1.01e+004	1.32e+003
	TVAC	−9.77e+003	7.92e+002
	DPSO	−1.38e+004	7.35e+002
100	SPSO	−1.81e+004	2.20e+003
	TVAC	−1.79e+004	1.51e+003
	DPSO	−2.72e+004	1.19e+003
200	SPSO	−3.13e+004	4.21e+003
	TVAC	−4.02e+004	4.36e+003
	DPSO	−5.51e+004	1.99e+003
300	SPSO	−4.34e+004	7.01e+003
	TVAC	−5.69e+004	3.51e+003
	DPSO	−7.99e+004	4.49e+003

Table 2
The comparison results of Rastrigin function

Dim.	Alg.	Mean	Std.
30	SPSO	1.99e+001	5.17e+000
	TVAC	1.60e+001	4.06e+000
	DPSO	6.40e+000	5.07e+000
50	SPSO	3.99e+001	7.93e+000
	TVAC	3.64e+001	6.52e+000
	DPSO	1.53e+001	5.58e+000
100	SPSO	9.37e+001	9.96e+000
	TVAC	8.81e+001	9.12e+000
	DPSO	4.14e+001	7.33e+000
200	SPSO	2.23e+002	1.74e+001
	TVAC	1.94e+002	3.08e+001
	DPSO	9.98e+001	1.14e+001
300	SPSO	3.63e+002	1.76e+001
	TVAC	2.66e+002	2.92e+001
	DPSO	2.12e+002	3.71e+001

Table 3
The comparison results of Ackley function

Dim.	Alg.	Mean	Std.
30	SPSO	7.59e–006	1.038e–005
	TVAC	6.51e–014	8.53e–014
	DPSO	4.78e–011	9.15e–011
50	SPSO	1.70e–004	1.28e–004
	TVAC	9.95e–005	1.73e–004
	DPSO	1.58e–008	1.78e–008
100	SPSO	3.31e–001	5.01e–001
	TVAC	4.69e–001	1.91e–001
	DPSO	3.68e–007	1.63e–007
200	SPSO	2.13e+000	2.19e–001
	TVAC	6.94e–001	4.08e–001
	DPSO	9.49e–007	4.07e–007
300	SPSO	3.15e+000	5.60e–001
	TVAC	7.66e–001	3.16e–001
	DPSO	1.59e–006	7.38e–007

Table 4
The comparison results of Penalized function

Dim.	Alg.	Mean	Std.
30	SPSO	7.38e–004	2.79e–003
	TVAC	7.93e–023	2.50e–022
	DPSO	5.16e–023	1.74e–022
50	SPSO	6.74e–003	5.44e–003
	TVAC	1.19e–002	3.03e–002
	DPSO	1.62e–017	3.93e–017
100	SPSO	2.90e+001	1.53e+001
	TVAC	3.77e–001	6.13e–001
	DPSO	8.24e–011	1.79e–010
200	SPSO	1.81e+003	1.74e+003
	TVAC	2.17e+000	1.61e+000
	DPSO	1.74e–010	1.07e–010
300	SPSO	1.47e+004	9.03e+003
	TVAC	3.73e+000	2.68e+000
	DPSO	4.03e–011	3.31e–010

Based on the above analysis, we can draw the following two conclusions:

(1) Dispersed particle swarm optimization is fit for solve high dimensional problem with strong linkages, especially for dimensionality is larger than 100.

(2) For numerical problem with weak linkages, DPSO can improve the performance than MPSO-TVAC and SPSO at least 20%.

5. Conclusion

This paper enlarges the action of social coefficient, and propose a new dispersed particle swarm optimization. In DPSO, the social coefficient uses a dispersed setting associated with the algorithm performance, and provides some probabilities to search around the personal historical best position. To our knowledge, this is the first report of using social coefficient to improve the convergent speed. The further research topic includes the other dispersed setting methods, and the applications.

Acknowledgement

This paper were supported by National Natural Science Foundation of China under Grant No. 60674104. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

References

[1] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.
[2] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, IV, 1995, pp. 1942–1948.
[3] He.Z. Wei, C. Yang, L. Gao, X. Yao, R.C. Eberhart, Y. Shi, Extracting rules from fuzzy neural network by particle swarm optimization, in: Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA, 1998.
[4] Z.H. Cui, J.C. Zeng, G.J. Sun, Adaptive velocity threshold particle swarm optimization, in: Proc. Int. Conf on Rought Sets and Knowledge Technology (RSKT 2006), Chongqing, China, in: Lecture Notes in Artificial Intelligence, vol. 4062, Springer, 2006, pp. 327–332.
[5] H.Y. Shen, X.Q. Peng, J.N. Wang, Z.K. Hu, A mountain clustering based on improved PSO algorithm, in: Advances in Natural Computation, First International Conference, ICNC 2005, Changsha, China, in: Lecture Notes in Computer Science, vol. 3612, Springer, 2005, pp. 477–481.
[6] Z.H. Cui, J.C. Zeng, G.J. Sun, Hybrid method to computing global minimizers combined with PSO and BPR, Chinese Journal of Electronic 15 (4A) (2006) 949–952.
[7] Z.H. Cui, J.C. Zeng, G.J. Sun, A fast particle swarm optimization, International Journal of Innovative Computing, Information and Control 2 (6) (2006) 1365–1380.
[8] S. Hassas, Using swarm intelligence for dynamic web content organizing, in: Proceedings of 2003 IEEE Swarm Intelligence Symposium, Indianapolis, USA, 2003, pp. 19–25.
[9] E. Ozcanand, C.K. Mohan, Particle swarm optimization: surfing the waves, in: Proceedings of IEEE Congress on Evolutionary Computation, 1999, pp. 1944–1999.
[10] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 240–255.
[11] G. Venter, Particle swarm optimization, in: Proceedings of 43rd AIAA/ASME/ASCE/AHS/ASC Structure, Structures Dynamics and Materials Conference, 2002, pp. 22–25.
[12] Y. Peng, X.Y. Peng, Z.Q. Liu, Statistic analysis on parameter efficiency of particle swarm optimization, Acta Electronica Sinica 32 (2) (2004) 209–213 (in Chinese).
[13] van den Bergh, An analysis of particle swarm optimizers, Ph Dissertation, Pretoria, University of Pretoria, 2001.
[14] X. Yao, Y. Liu, G.M. Lin, Evolutionary programming made faster, IEEE Trans. Evolut. Comput. 3 (1999) 82–102.