A synergetic particle swarm optimization algorithm-DHPSO

Jinchao.Guo
skyzzuli@126.com
Department of control science & engineering
huazhong University of science & technology
Wuhan, HuBei,430074

Guangzhao.Cui
cgzh2008@163.com
Key lab of informationed apparatus
Zhengzhou university of Light industry
Zhengzhou, HeNan. 450002

Abstract

To improve the design-quality of robotics, more and more artificial intelligence and evolutionary algorithm were introduced into the design procedure, such as routing-programming, data-fusion, vision-processing, and so on. These new methods prompt the development of robotics greatly. An improved Particle Swarm Optimization (PSO) method, Dual-hierarchy Particle Swarm Optimization algorithm (DHPSO) was proposed in this paper. Compared with the regular particle swarm optimizer (RPSO), DHPSO adopts dual hierarchy structure. In the bottom layer, several particle groups try to find out current optimal solutions in the multi dimension searching space respectively. In the top layer, one group particles chase the global optima. The proposed method is tested on three benchmark functions. All simulation results show the proposed method is better than the regular PSO in speed and precision performance.

Keywords: Robotics, PSO, DHPSO, Hierarchy structure

1 introduction

Nowadays swarm algorithms and other intelligent algorithms are used widely in many fields. We have used Particle Swarm Optimization (PSO) algorithm in route programming of robot design successfully and made some improvement of this algorithm on its structure. More will be discussed about Particle Swarm Optimization algorithm below. Just like a wealth of heuristic algorithms such as genetic, evolution, and simulated annealing. Particle Swarm Optimization algorithm is a new entrant to the family of evolutionary algorithms originally introduced by Kennedy and Eberhart in 1995[1][2]. As parameters in PSO algorithm is fewer contrast to other evolution algorithm, moreover PSO algorithm is easy to realize and has a good performance in searching for the optima in real value searching space. Researchers and experts pay a great deal of attention to it in the last decades. Though PSO has so many merits, it's easy to be trapped in local optima, and other drawbacks such as premature, low precision, and non-convergence. In order to overcome these demerits of PSO algorithm, many researchers make lots of contributions, and give out many modified algorithms based on the regular PSO. Junjun Li and Xinhua Wang propose a refined PSO algorithm based on simulated annealing[6]; Qianli Zhang, Xing Li and Quang-AHN TRAN introduce mutation operator to the regular PSO algorithm[7]; Chnming Yang and Dan Simon made Each particle learn from its previous worst position and its group's previous worst position, and give out a novel method to program PSO algorithm[8]; Jang-Ho Seo, Chang-Hwan Im, Chang-Geum Heo and Jae-Kwang use N groups of particle swarm to do the optimization, and get a new algorithm MGPSO[9]. Most of these algorithms put emphasis on parameter selection or combination with other evolution algorithms to get new derivations. This is difficult to solve the innate flaw of PSO algorithm-Local minima trapping, furthermore, enhance the complexity. Under these premises, we proposed dual hierarchy PSO algorithm, it holds the concepts of the regular PSO and tries to keep balance between "exploration" and "exploitation" between different hierarchies. Compared with the regular PSO algorithm, DHPSO improved searching speed and enhanced the global optima searching ability. The paper is organized as follows. In Section II, we introduce the concept of regular PSO and the method used in the study DHPSO, Section III outlines the experimental setup, parameter settings, and benchmark functions used. The experimental results are presented in Section IV. Finally, Section V contains a discussion of the experimental results.

2 The regular PSO and Proposed algorithm

2.1 The regular PSO

PSO algorithm was developed by Kenney and Eberhart to model birds flocking and fish schooling for food, in which particles representing the candidate solutions to the problem in a multidimentiona1 search space. Every particle has a position vector $\vec{x}$ encoding a candidate solution to the problem and a velocity vector $\vec{v}$. Moreover, each particle contains a small memory that stores its own best position seen so far p_{id} and a global best position p_{gd} obtained through communication with its neighbor particles. The position and velocity vector of each particle are updated every iteration, and this is done respectively according to equation 1 and 2 as below, where ω is known as the inertia weight as described in[3][4]. The parameters c_1 and c_2 are set to constant values, which are normally given as 2 , and rand is randomly generated value between 0 and 1. $i \in (1, 2 \cdots N)$ and $N \in (1, 2 \cdots D)$. N and D denote particle number and searching space dimension respectively. Inertia weight ω is adjusted with interaction according to equation 3.

$$\begin{aligned} v_{i+1,d} &= \omega \cdot v_{id} + c_1 \cdot \text{rand}\left(p_{id} - x_{id}\right) + \\ &+ c_2 \cdot \text{rand}\left(p_{gd} - x_{id}\right) \end{aligned} \tag{1}$$

$$x_{i+1,d} = x_{id} + v_{i+1,d} \tag{2}$$

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{\text{Gen}_{max}} \cdot \text{i} \tag{3}$$

2.2 Proposed algorithm

Some researchers give out the concepts of ”swarm fitness covariance” and ”degree of convergence” and try to make criteria on which to investigate the convergence of algorithm. All these make contribution to meliorate regular PSO in a way, However, these methods can’t improve PSO algorithm in essence, because they just try to enhance the global optimization searching rate by repeat of program running in a sense. Some existing methods adopt operators of genetic algorithm or evolutionary strategy, such as crossover, mutation, and sharing. These methods can modify regular PSO algorithm in some ways, but improve complexity considerably at the same time. All these make modified PSO algorithm not easy to realize. If groups are used for DHPSO, the computational cost increases only in the order of $O\left(N\right)$ compared with one-group PSO because each group follows the basic concepts of conventional PSO algorithm, whereas roughly $O\left(N^2\right)$ increments is required for GA crossover computations if an larger number of populations is used in GA. In DHPSO algorithm, The velocity of the bottom layer updating adopts equation 1 to guarantee a good local ”exploration” performance, and in the top one, velocity updates according to equation 4–the local updating formula, to gain a better convergence speed.

$$v_{i+1} = \omega \cdot v_i + c_2 \cdot rand\left(p_{gd} - x_{id}\right) \tag{4}$$

3 Preparation of experiment and benchmark function

c_1 and c_2 are set to 2, the total population N of each algorithm will be set to equal, in experiment it is 1000. More seeds will be required when problem is complex. and ω_{max} and ω_{min} are set to 0.9 and 0.4, vector $\vec{x}$ belongs to a scope of[-50 50] so that the search process will not last too long, vector $\vec{v}$ is set to be [-20 20]. For DHPSO, At different hierarchy the particle number in each group can be different, but the total number N–sum of particle number of different hierarchy should be the same with that of regular PSO algorithm, just to keep same complexity. In experiment, we set two groups in the bottom layer of DHPSO, and each group has 150 particles, so the top layer number is 700 obviously. The dimension number D is set to 2 and 10 respectively. Comparison functions adopted here are three benchmark functions used by many researchers. They are the Griewank, Rastrigrin and Rosebrock functions. The definitions of these functions are presented below:

$$f_1\left(x\right) = \frac{1}{4000} \cdot \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{5}$$

$$f_2\left(x\right) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos 2\pi x_i + 10\right) \tag{6}$$

$$f_3\left(x\right) = \sum_{i=1}^{n} \left(100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right) \tag{7}$$

4 Simulation result comparison

Each test repeats 10 times and each iteration runs 100 generations. We list out the different minima of the benchmark functions that we got using RPSO and DHPSO method. The preliminary conditions are specified in the table list.

Fig.1 to Fig.6 show the convergence result of DHPSO and RPSO. The benchmark function used and the particular conditions of each figure are specified in the figure caption.

Table 1: the mean optimization result of RPSO and DHPSO with D =2 and N=1000

funcion	RPSO	DHPSO
f_1	0.0000	0
f_2	0.0000	0
f_3	0.0000	0

Table 2: The mean optimization result of RPSO and DHPSO with D =10 and N=1000

funcion	RPSO	DHPSO
f_1	0.1076	0.1038
f_2	22.2080	18.6500
f_3	0.0000	0.0000

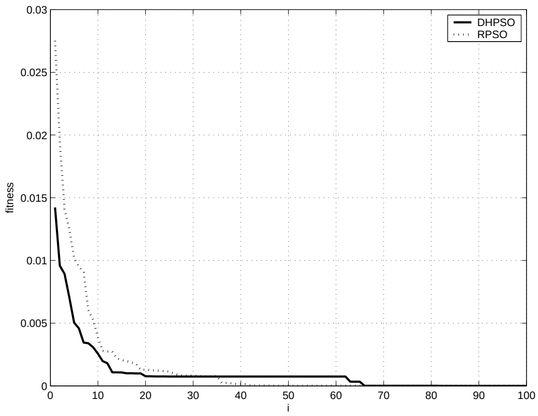

Fig. 1: Griewank, N=1000, D=2

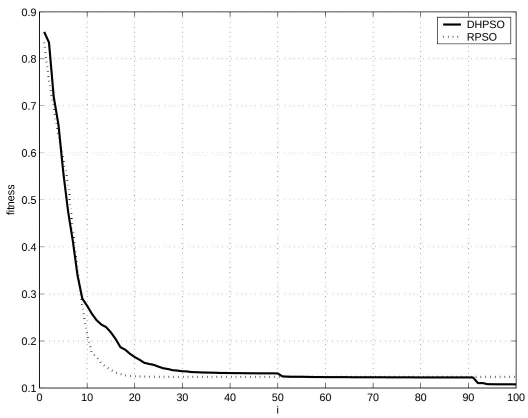

Fig. 2: Griewank, N=1000, D=10

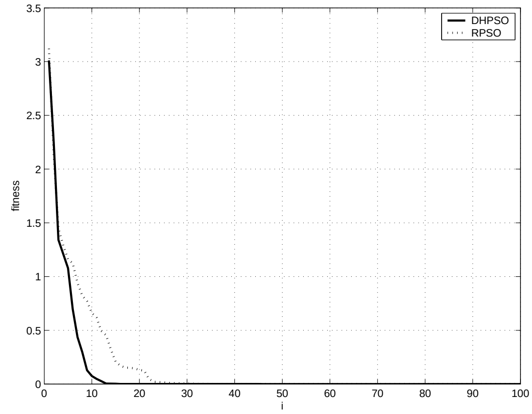

Fig. 3: Rastrigrin, N=1000, D=2

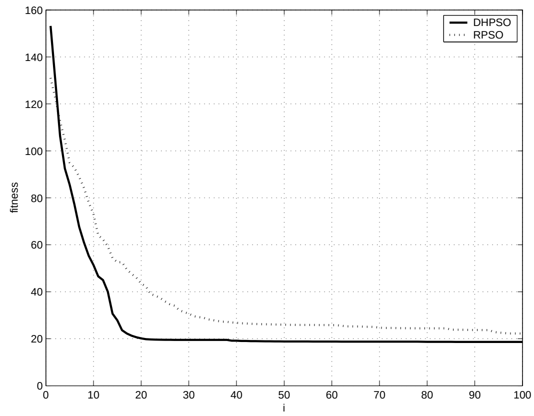

Fig. 4: Rastrigrin, N=1000, D=10

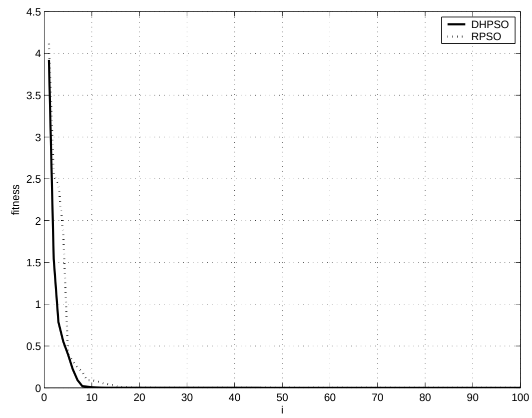

Fig. 5: Rosebrock, N=1000, D=2

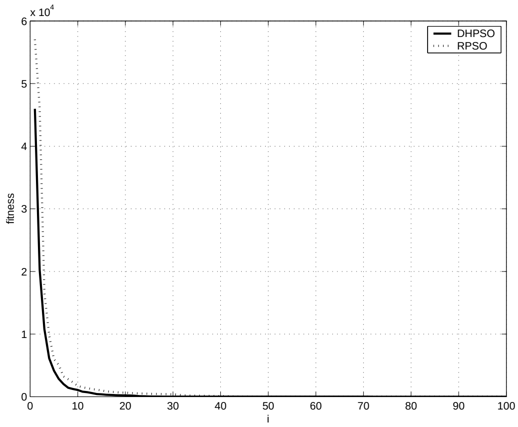

Fig. 6: Rosebrock, N=1000, D=10

Because vector $\vec{x}$ and vector $\vec{v}$ are generated randomly, the initial value may be different for the two algorithms in the beginning. However, the two algorithms still can be compared, for they have the same variation iteration and same complexity degree. From Fig.1 to Fig.6 above, we can get that DHPSO method has a good precision than RPSO method, and as well as the convergence speed.

5 Conclusion

From the simulation results, we can see DHPSO has a better performance than the RPSO algorithm, but these experiments are executed under a particular cases and narrow settings, so the conclusion is not comprehensive and definite. For further research, more work can be put into the following aspects:

1. The relation ship between inertia weight and convergence of algorithm.
2. How to improve global searching ability of algorithm.
3. Seeking the theory support of PSO algorithm.
4. Combining other heuristic algorithm and applying PSO to more engineer use.

References

[1] R. C. Eberhart and J. Kennedy, "A New Optimizer Using Particles Swarm Theory," in Proc. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39-43.

[2] R. C. Eberhart and J. Kennedy, "Particles Swarm Optimization," in Proc. IEEE International Conference on Neural Network, Perth, Australia, 1995.

[3] Shi, Y.H., Eherhart, R.C., (1998). A modified particle swarm optimizer. Proceedings of the IEEE International Conference on Evolutionary Computation., Anchorage, Alaska, May 4-9, 1998.

[4] Shi, Y.H, Eberhart, R.C., (1999). Empirical study of particle swarm optimization, 1999 Congress on Evolutionary Computation, Washington DC, USA, July 6-9,1999.

[5] S.L.HO, Shiyou Yang, Guangzheng Ni ang H.C.Wang, "A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices," IEEE, Tran. Magn, Vol.42, No.4, 2006

[6] Junjun Li and Xihuai Wang, "A modified particle swarm optimization algorithm," in Proc. IEEE world congress on intelligent control and Automation, Hangzhou, China., 2004 ,pp.354-356.

[7] Qianli Zhang, Xing Li and Guang-AHN.TRAN, "A modified particle swarm optimization algorithm," in Proc, IEEE International conference on Machine learning and cybernetics, Guangzhou, China, 2005.pp.2993-2995.

[8] Chuming Yang and Dan Simon, "A new particle swarm optimization technique," in Proc. IEEE, International conference on systems engineering, August,2005,pp.164-169.

[9] Jang-Ho Seo,Chang-Hwan Im, Chang-Geum Heoand Jae-Kwang Kim, "multimodal function optimization based on particle swarm optimization," IEEE, Trans. Magn, vol. 42, No. 4, April 2006,pp. 1095 - 1098.