International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 26559-26564

http://www.ripublication.com

A Hybrid Dynamic Multi-Swarm Particle Optimizer With Cooperative Learning Strategy

S.Bhaskaran[1] and P.Swaminathan [2]
School of Computing, SASTRA University, Thanjavur India

Abstract

In this paper, three variations of particle swarm optimizers PSO, DMS-PSO, and new cooperative learning strategy DMS-PSO-CS are discussed. This paper emphasizes cooperative learning strategy that applies learning to be used more effectively and to generate better quality. In this paper, for effective and optimal search each sub-swarm is used so that each dimension of the two inefficient particles can learn from the better particle. The selection methodology used is tournament method. The results show that DMS-PSO-CS has a clear cut advantage over the other two s in comparison with DMS-PSO and standard PSO.

Keywords: Particle swarm optimizer, hybrid multi-swarm particle optimizer, Cooperative learning strategy.

Introduction

The PSO is a computational evolution technique that is mimicked from the natural swarm behaviors of birds flocking and fish schooling [1]. A particle in PSO uses the information of its historical best position and its neighborhood's best position to adjust its flying velocity to search for the global optimum in the solution space [4]. The PSO algorithm is not very efficient when solving complex problems because it is easy to be trapped into local optima. The easiness of getting trapped into local optima is caused by that PSO does not sufficiently utilize its population's search information to guide the search direction. Thus PSO has difficulty in solving complex problems [2] [6] [7]. The original PSO is a global version PSO (GPSO) where all the particles are attracted by the same globally best particle and the swarm has tendency to fast converge to the current globally best point. Since GPSO uses only the search information of the globally best particle to guide the search direction, it may lead to premature convergence due to the lack of diversity. To get the diversity niching methods are used in genetic algorithm. Therefore, GPSO is not very efficient when solving complex multimodal functions as the particles cannot efficiently use search information of the whole swarm to find out the global optimum[3]. As the PSO has become more attractive and has been utilized in lots of real world applications, many researchers have been working to improve the algorithm performance and various PSOs have been proposed. Some researches focused on the parameter studies such as the inertia weight and the acceleration coefficients. Also, some researchers concentrated on combining PSO with other evolutionary operators and techniques to improve PSO's performance. Inspired by biological mechanisms, niche and speciation technology are introduced into PSO to prevent particles to be too close to each other so that PSO can locate as many optimal solutions as possible. On the other hand, different topology structures, such as the star, ring, pyramid, and von Neumann structures have been studied to improve the algorithms performance[8][5]. PSO variants enhanced by orthogonal learning strategy, neighborhood search, centripetal strategy, multi-layer search strategy, self-adaptive strategy, and intermediate disturbance strategyhave attracted great attentions[1][3].

Review of PSO and DMS-PSO

Standard PSO

The efficient and best solution to the optimization problem can be specified as a point and can be represented in D-dimensional space for an optimization problem has D variables to optimize. Each particle has a velocity vector to deter-mine its direction and a fitness value to measure its corresponding optimization state[7]. The position and velocity in D-dimensional search space are adjusted according to the current optimal particle. The process can be converted into a mathematical problem as follows. Suppose that sz particles are used to search the solution. The ith particle in D-dimensional space is represented as $x_i=(x^1_i, x^2_i, \cdot\cdot\cdot, x^d_i, \cdot\cdot\cdot, x^D_i)$, where $x^d_i \in [x_{min}, x_{max}]$, $d \in [1, D]$.

The velocity corresponding to the ith particle is $vi= (v^1_i, v^2_i, \cdot\cdot\cdot, v^d_i, \cdot\cdot\cdot, v^D_i)$,where$v^d_i \in [v_{min}, v_{max}]$.

The velocity and location update strategyof the ith particle are given below:

$$v^d_i \leftarrow v^d_i + c1 \cdot rand1_i^d \cdot (pbest^d_i - x^d_i) + c2 \cdot rand2_i^d \cdot (gbest^d_i - x^d_i) \quad (1)$$

$$x^d_i = x^d_i + v^d_i \quad (2)$$

where c1and c2are the acceleration constants. c1 represents the weight that the ith particle tracks its own historical optimum value $pbest_i$. Figuratively speaking, it shows the understanding of itself. Similarly, c2 represents the weight that the ith particle tracks the whole group's optimum value gbest. All particles use the same values c1and c2. $pbest_i$ and gbest are updated all the time according to each particle's fitness value. $rand1^d_i$and $rand2^d_i$ are two random numbers in [0, 1]. To control the flying velocity, an inertia weight or a constriction factor is introduced It is modified to be Eq. (3)

$$v^d_i \leftarrow w \cdot v^d_i + c1 \cdot rand1_i^d \cdot (pbest^d_i - x^d_i) + c2 \cdot rand2_i^d \cdot (gbest^d_i - x^d_i) \quad (3)$$

where w usually decreases linearly from 0.9 to 0.4 during the iterative process

Substantially, PSO is divided into two versions. The above formula is global PSO, another version is local PSO[9]. For local PSO, eachparticle adjusts its position and velocity according to its historical best position $pbest_i$ and the best position achieved so far from its group $lbest_i$. The velocity update strategy is described as follows:

$$v^d_i \leftarrow w \cdot v^d_i + c1 \cdot rand1^d_i \cdot (pbest^d_i - x^d_i) + c2 \cdot rand2^d_i \cdot (lbest^d_i - x^d_i) \quad (4)$$

DMS-PSO

DMS-PSO is a new version of PSO. This method search for better position using their own members of population. Since the sub-swarms are dynamic and they are regrouped frequently by using a regrouping schedule.. In this way, the search space of each population is expanded and unique diverse solutions are to be found in the next iteration. A very small population size (e.g., 3∼5) is enough when solving relatively complex problems, which is also one of its significant features. To illustrate this algorithm deeply, detailed description is presented. Take the population size sz=9 X = (x1, x2,· · · , xi, · · · , x9) and divide them into three sub-swarms L = (L1, L2, L3)randomly. Then each particle works within the sub-swarm where it stays. Its position is updated through Eq. (4) and (2). The new population may converge to a local optimum during this period. Then the pattern is broken and sub-swarms are recombined. Based on this, particles start updating again and the process is repeated.

The regrouping period R is a key parameter which has a great influence on the optimization results. If R is set too small, then there will be less number of generations for each population to successfully complete a search. If R is set too big, function evaluations will be exhausted and wasted as a result sub-swarms will not further succeed. Hence it should neither be too small nor too large. The whole iterative process is divided into two phases. The top ninety percent of all iterations run as the above description. The process is mostly used to conduct an extensive search. The remaining percent of iterations run as the global version PSO and can be explained as a targeted search procedure. The pseudo-code of DMS-PSO has been given below in Algorithm 1.

Algorithm 1. The pseudo-code of DMS-PSO.

Initialize:
sz: size of the whole population;
numregion: number of sub-swarms;
R: regrouping period;
i_ max: max iterations;
Initializeeach particle’s position x^d_i and velocity v^d_i and divide the particles into numregion sub-swarms;

Iterate:
1: for iter=1:0.9*i_max do
2: Update each particle;
3: if mod(iter,R)=0then
4: Rearrange the sub-swarms randomly;

5: end if
6: end for
7: for iter=0.9*i_max:i_maxdo
8: Update each particle using global version PSO;
9: end

DMS-PSO With Cooperative Learning Strategy

Although DMS-PSO can solve multimodal problems efficiently, there still exist some drawbacks. In DMS-PSO, each particle in a sub-swarm only learns from its pbest and lbest. Information among different sub-swarms can't be exchanged until the population is regrouped due to the deficiency of cooperative learning. DMS-PSO achieves great improvement on global exploration but lacks local exploitation. Aiming at this drawback, a new strategy of multi-swarm cooperative learning strategy is used with DMS-PSO, which is used to exchange information among different sub-swarms before the regrouping operation. In this way, the collaborative learning among sub-swarms is enhanced and the balance between the global exploration and the local exploitation can be achieved. The cooperative learning procedure is described as follows:

1. For each sub-swarm, we sort the fitness values of the particles and select the two worst particles to be updated
2. For each particle's each dimension, we select two sub-swarms randomly out of the whole groups which include the sub-swarm where the particle to be updated stays.
3. We compare the fitness values of the two sub-swarms lbests and select the better one.
4. We use the winners lbest of the particle to be updated.

The pseudo-code of the cooperative learning strategy is given in Algorithm 2.

Algorithm 2. The pseudo-code of the cooperative learning strategy

Sort:

Sort fitness values of each groups particles; Select the worst two particles $lworst^k_i$ k = 1, 2 and the best one particle lbesti of each sub-swarm

Replace:

1: for each $lworst^k_i$ do
2: for each dimensional $lworst^k_i(d)$ do
3: Select two local best particles lbest1(d), lbest2(d) randomly and compare their fitness values $f(lbest_1(d))$, $f(lbest_2(d))$;
4: if $f(lbest_1(d)) < f(lbest_2(d))$then
5: $lworst^k_i(d) = lbest_1(d)$;
6: else
7: if $f(lbest_1(d)) > f(lbest_2(d))$then
8: $lworst^k_i = lbest_2(d)$;
9: end

Experiments and Results

In this section, experiments are conducted. Three functions are used in the comparison 1. Sphere 2. Elliptic 3.Ackley. Each function is repeated 20 times independently. The proposed DMS-PSO-CS algorithm and six other PSO variants are implemented in this section for comparison purpose. They are detailed in Table 1.

Table 1:

Parameter settings of the seven PSOs.

Algorithm	Parameters Setting
GPSO	w: 0.9~0.4, $c_1 = c_2 = 2.0$, $v_{max} = 0.2 \times$ Range
PSO-cf	$c_1 = 3.0$, $c_2 = 2.0$, $v_{max} = 0.2 \times$ Range
CLPSO	w: 0.9~0.4, $c_1 = c_2 = 1.49445$, $v_{max} = 0.2 \times$ Range, Refreshing gap = 7
SPSO	$w = 1/(2^*\log(2))$, $c_1 = c_2 = 0.5 + \log(2)$, $v_{max} = 0.2 \times$ Range
CPSO	w: 0.9~0.4, $c_1 = c_2 = 1.49445$, $v_{max} = 0.2 \times$ Range
DMS-PSO	w: 0.729, $c_1 = c_2 = 1.49445$, $v_{max} = 0.2 \times$ Range, Regrouping period = 5
DMS-PSO-CLS	w_1: 0.9~0.4, $w_2 = 0.2$, $c_1 = c_2 = 2$, $v_{max} = 0.2 \times$ Range, Regrouping period = 5

In order to achieve a better performance for DMS-PSO-CS, w is typically set to 0.8 reducing linearly to 0.4. Then a greater exploration can be achieved at the beginning and later the emphasis is shifted to a greater exploitation. In DMS-PSO-CS, each sub-swarm contains 3 individuals.

The following major conclusions of DMS-PSO-CS can be achieved as below:

Table 2:

The computational time of PSOs.

Time (s)	GPSO	PSO-cf	CLPSO	SPSO	CPSO	DMS-PSO	DMS-PSO-CLS
f_1	0.194	0.192	0.237	0.203	0.343	0.339	0.600
f_2	0.157	0.154	0.182	0.168	0.322	0.291	0.548
f_3	0.191	0.192	0.210	0.203	0.368	0.343	0.633

Conclusion

In this article the comparison of PSO techniques help to improve the performance of DMS-PSO. The new strategy makes particles to learn from and find the global optimum more easily. From the results, it can be observed that the cooperative learning strategy enables DMS-PSO-CS to take full advantage of the shared information and effectively improve the convergence speed and accuracy of DMS-PSO. Based on the results, it can be concluded that DMS-PSO-CS has superior features both in high quality of the solution and robustness of the results. Future work will be focused on the study of multi-swarm and the better cooperation mechanism between groups. In addition, applying the proposed optimization algorithms to complex practical engineering problems is also our future work.

References

[1] J. Kennedy, R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, 2001.

[2] T.K. Das, G.K. Venayagamoorthy, U.O. Aliyu, Bio-inspired algorithms for the design of multiple optimal power system stabilizers: SPPSO and BFA, IEEE Trans. Ind. Appl. 44 (5) (2008) 1445–1457.

[3] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inf. Sci. 209 (2012) 16–36.

[4] M.R. Chen, X. Li, X. Zhang, Y.Z. Lu, A novel particle swarm optimizer hybridized with extremal optimization, Appl. Soft Comput. 10 (2) (2010) 367–373.

[5] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings ofthe IEEE World Congress on Evolutionary Computational Intelligence, 1998, pp.69–73.

[6] J. Kennedy, Stereotyping: improving particle swarm performance with cluster analysis, in: Proceedings of the IEEE Congress on EvolutionaryComputation, 2000, pp. 1507–1512.

[7] 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, 1995, pp. 39–43.[2]

[8] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[9] T. Blackwell, A study of collapse in bare bones particle swarm optimization, IEEE Trans. Evol. Comput. 16 (3) (2012) 354–372.[4] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 1999, pp.1945–1950.