The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

Competition-based Particle Swarm Optimizer for Solving Numerical Optimization Problems

Sheng-Ta Hsieh and Bo-Han Chen

Department of Communication Engineering, Oriental Institute of Technology, Taiwan
(Tel: +886-2-7 738-0145 Ext. 2325, Fax: +886-2-7738-7411)

fo013@mail.oit.edu.tw

Abstract: In this paper, a new variant of particle swarm optimizer is proposed for solving numerical optimization problems. The main difference between proposed method and common PSO is the swarm structure. In general, the PSO has only one swam and each particle of the swarm will share their information for guiding other particles toward to potential solution space. The proposed method is to separate swarm into two sub-swarm. The size of two sub-swarms will be adjusted according to their performance. It can drive increased the diversity of the particles and prevent particle to fall into the local optimum. For testing the performance of proposed method, fifteen of CEC 2005 test functions were selected for experiments. From the result, it can be observed that the proposed method performs better than SPSO2011.

Keywords: computational intelligence, optimization, particle swarm optimizer, population.

1 INTRODUCTION

In last 3 decades, evolutionary algorithms were applied for solving variety of real world problems. Since 1975, John Holland proposed the first evolutionary based optimizer, genetic algorithm (GA)[1]. More and more optimizer was then proposed, such as, ant colony optimization (ACO)[2], particle swarm optimization (PSO)[3], differential evolution (DE)[4], and artificial bee colony algorithm (ABC)[5], etc. The main feature of PSO is it has simply equations to generate new moving vector and has and very few parameters to be set. The Particle swarm optimization (PSO) was introduced by Kennedy and Eberhart in 1995. The PSO Simulated foraging behavior of birds or fishes to solve optimization problems. Particles of the swarm will move in a D-dimensional space according to new moving vector. The positions of particles represent potential solutions. Each particle will have chance to share their better experience (global best solution, *Gbest*) for guiding other particle toward to unsearched or potential solution space. The global best solution is selected from the best performance of the personal best solutions (*Pbest*) of all particles. Both of the *Gbest* and *Pbest* will then guide particles' for searching optimal solution.

In last decade, more and more variants of PSO have been proposed to improve PSO's efficiency and make it much robust. Omran *et al.* introduced a hybrid of particle swarm optimization and differential evolution [6]. Chen and Yeh presented a new search strategy [7]. They keep only personal best position for providing moving vector but no global best position. It will make vector update much simply, and reduce computational consumption. But is will also decrease particles' ability for deep search. Some methods were focus on parameters adjustment. Due to different setting of parameters will affect optimizer's performance directly. In [8] Bratton and Kennedy proposed two kinds of topology type which are *Gbest* and *Lbest* topology. The *Gbest* type is for solving uni-modal problems; and *Lbest* type is much suitable for solving multi-modal problems. Ghosh *et al.* proposed hierarchical dynamic neighborhood PSO [9]. It will arrange particles for define their neighborhood under a dynamic hierarchical structure. Neighborhoods of a particle will changes constantly according to their solution quality. The changing of the arrangement (relationship of particle's neighborhoods) can preserve particle's diversity for easier finding optimal solution.

In general, the main advantage of PSO is fast convergence, but the weakness is easier to fall into local optimum. Due to particles will toward around global optimal solution. In order to overcome this situation, Blackwell and Branke proposed multi-swarm PSO[10], which divided population into several sub-swarms. Each swarm has its own parameters and will perform evolution independently. After a few generations, sub-swarm will perform information sharing. Since there are more than one global optimal solution, It can avoid particles form fall into local optimum.

In order to further improve PSO's performance, a competition-based particle swarm optimizer is proposed. The population is separated as two sub-swarms. After several generations, performance of two sub-swarms will take into comparison. The swarm size will be adjusted according competition results.

The paper is organized as follows: In Section II, a simply introduce of standard PSO will be described. Section III presents the detail of proposed method. Section IV presents the experiments including test functions and parameter settings. Finally, the conclusion is described in Section V.

2 PARTICLE SWARM OPTIMIZATION

2.1. Original PSO

Since the original PSO proposed in 1995[3], researchers are focus on improve its solution searching ability and applied it on various applications. In order to extend its ability of exploration and exploitation, Shi and Eberhart [11] introduced *inertia weight* into the original PSO, for enhancing particles' search abilities and drive particle keep closer on potential solution space. The modified PSO (with inertia weight) will make PSO much robust and increase its searching performance.

In PSO, particle's movement are represented by position vector *x* and velocity *v*. The new velocity for particle *i* is generated by equation (1). Then, the *i*th particle's new position will be updated by equation (2).

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times r_1 \times (pb_{id}(t) - x_{id}(t)) + c_2 \times r_2 \times (gb_{id}(t) - x_{id}(t) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

Where, *d* denotes dimension, *gb* is the global best particle, and *pb* is the personal best particle. The *w* represents inertia weight, c_1 and c_2 are acceleration coefficients, r_1 and r_2 are random number between [0, 1].

The velocity update equation is consists of three parts, including the previous velocity, personal experience and social experience. They are controlled by inertia weight and two acceleration coefficients respectively.

3 COMPRTITION-BASED PARTICLE SWARM OPTIMIZATION

In this paper, competition-based particle swarm optimizer is proposed for solving numerical optimization. For the proposed method, the population is divided into two sub-swarms. Either of sub-swarm will apply different parameters setting.

3.1 Sub-Swarm Setting

The velocity update of proposed is listed as follows.

Velocity update for Sub-swarm1:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times r_1 \times (pb_{id}(t) - x_{id}(t)) + c_2 \times r_2 \times (lb^1_{id}(t) - x_{id}(t)) \quad (3)$$

Velocity update for Sub-swarm2:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times r_1 \times (pb_{id}(t) - x_{id}(t)) + c_2 \times r_2 \times (lb^2_{id}(t) - rpb_{id}(t)) \quad (4)$$

and the inertia weight is assigned according following equation.

$$w = (w_1 - w_2) \times \frac{(MAX\_FEs - FEs)}{MAX\_FEs} + w_2 \quad (5)$$

Where, *MAX_FEs* is the maximum fitness evaluations, the c_1 and c_2 are acceleration coefficients, and r_1 and r_2 are random number between $[0, 1]$.The lb^s is the local best solution of s_{th} sub-swarm, and *pb* is the personal best particle. The *rpb* is the random selected neighborhood *pb*, and *rpb* is not equal the current *pb*. The w_1 and w_2 denotes the lower and upper bound of inertia weight. The *w* for sub-swarm1 will decrease linearly, and the *w* for s ub-swarm2 is generated randomly. The range of two sub-swarms are set as [0.4, 0.9] and [0.45, 0.75] respectively.

3.2 Information Exchange and Sub-swarm Size Adjustment

In general, each sub-swarm will perform its own solution search process independently. After several generations, the sub-swarm with better performance will share its own *gbest* information to other sub-swarms. The pseudo code of information exchange is given as follows:

```
If   lbf^1 < lbf^2
     count = count + 1;
else
     count = count - 1;
End

If   count == 10
     lb^2 = lb^1;
     count = 0;
     regrouping = regrouping + 1;
elseif
     count == -10
     lb^1 = lb^2;
     count = 0;
     regrouping = regrouping - 1;
End
```

*lbf denoted fitness value of local best particle.

In order to provide better sub-swarm has much resource to keep finding better solutions, the sub-swarm size will be adjustment according its performance. Thus, after several generations, the better sub-swarm will have a new joined particle and the worse sub-swarm will reduce one. However, in order to keep sub-swarm's basic ability and to prevent all the particles of worse swarm be removed. The minimum sub-swarm size is set 10. The pseudo code of sub-swarm size adjustment is given as follows:

```
If regrouping == 20
    If    np2  > 10
    {regrouping sub-swarm}
    End
    regrouping =  0;
Elseif    regrouping == -20
    If    np1 > 10
    {regrouping sub-swarm}
    End
    regrouping =  0;
End
```

3.3 Flowchart of the CSPSO

The procedure of the CSPSO is given as follows:

Step 1: Initial population.

Step 2: Divide population into two sub-swarms.

Step 3: Calculate Fitness value for each sub-swarm.

Step 4: Perform solution search by two sub-swarms.

Step 5: Update each swarm local best particle(*lb*) and personal best particle(*pb*).

Step 6: Determine condition for information exchange (local best position).

Step 7: Determine condition for regrouping sub-swarms.

Step 8: Stop evolution if meet the stop condition, else jump to step 4.

TABLE 1. TEST FUNCTIONS

f	Test Functions
f_1	Shifted Sphere Function (f_1)
f_2	Shifted Schwefel's Problem 1.2 with Noise in Fitness (f_4)
f_3	Schwefel's Problem 2.6 with Global Optimum on Bounds (f_5)
f_4	Shifted Rosenbrock's Function (f_6)
f_5	Shifted Rotated Griewank's Function without Bounds (f_7)
f_6	Shifted Rastrigin's Function (f_9)
f_7	Shifted Rotated Weierstrass Function (f_{11})
f_8	Schwefel's Problem 2.13 (f_{12})
f_9	Expanded Extended Griewank's + Rosenbrock's Function (f_{13})
f_{10}	Hybrid Composition Function 1 (f_{15})
f_{11}	Rotated Hybrid Composition Function 2 (f_{18})
f_{12}	Rotated Hybrid Composition Function 2 with a Narrow Basin for the Global Optimum (f_{19})
f_{13}	Rotated Hybrid Composition Function 2 with the Global Optimum on the Bounds (f_{20})
f_{14}	Rotated Hybrid Composition Function 3 with High Condition Number Matrix (f_{22})
f_{15}	Rotated Hybrid Composition Function 4 without Bounds (f_{25})

4 EXPERIMENTAL RESULTS

In order to test proposed method and compare it to standard particle swarm optimization (SPSO2011) [12], fifteen test functions of the CEC 2005 [13], which includes included uni-model functions (f_1 ~ f_3), multi-model functions (f_4 ~ f_8), expanded functions (f_9), and hybrid composition functions (f_{10} ~ f_{15}), are listed in Table 1. The global optimum, initialization and search range of the 18 test functions are listed in Table 2. All algorithms are implemented by MATLAB 2010a and are executed on platform with Intel Core i3-2120 and 4GB RAM.

TABLE 2. THE GLOBAL OPTIMUM, INITIALIZATION RANGE AND SEARCH RANGE OF TEST FUNCTIONS.

f	Global Optimum	Initialization	Search Range
f_1	0	$[-100, 100]^D$	$[-100, 100]^D$
f_2	0	$[-100, 100]^D$	$[-100, 100]^D$
f_3	0	$[-100, 100]^D$	$[-100, 100]^D$
f_4	0	$[-100, 100]^D$	$[-100, 100]^D$
f_5	0	$[-100, 100]^D$	$[-100, 100]^D$
f_6	0	$[-32, 32]^D$	$[-32, 32]^D$
f_7	0	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$
f_8	0	$[-100, 100]^D$	$[-100, 100]^D$
f_9	0	$[-3,1]^D$	$[-3, 1]^D$
f_{10}	0	$[-5,5]^D$	$[-5, 5]^D$
f_{11}	0	$[-5, 5]^D$	$[-5, 5]^D$
f_{12}	0	$[-5, 5]^D$	$[-5, 5]^D$
f_{13}	0	$[-5, 5]^D$	$[-5, 5]^D$
f_{14}	0	$[-5, 5]^D$	$[-5, 5]^D$
f_{15}	0	$[-2, 5]^D$	$[-2, 5]^D$

All the test functions are set as 30-dimension and executed for 25 times. Both the population size of SPSO

2011 and proposed method are set as 40, and the maximum fitness evaluations (FEs) are set 300,000. The details of related parameters are listed in Table 3.

TABLE 3. PARAMETER SETTINGS

Algorithm	**Proposed Method**	**SPSO 2011**
Inertia weight	[0.4,0.9] for sub-swarm 1 [0.45,0.75] for sub-swarm 2	0.7213
Population size	40	
Dimensions	30	
c_1, c_2	1.1931	
Max FEs	300,000	
Run	25	

The experimental results are listed in TABLE 4 which includes mean and standard deviation of 15 selected test functions of 25 independent runs. The better performance of two PSO variants is shown in bold. From the results, it can be observed that the proposed method perform better in most test functions, especially in the hybrid problems.

TABLE 4. EXPERIMENT RESULTS FOR 30-D PROBLEMS

Algorithms / Functions	**SPSO 2011**	**Proposed method**
f_1	1.3717e-027 (1.7744e-028)	**2.0195e-030** **(9.8934e-030)**
f_2	**6.0104e+001** **(3.4593e+001)**	1.0867e+002 (1.4697e+002)
f_3	4.7958e+003 (8.2740e+002)	**3.8486e+003** **(1.8659e+003)**
f_4	3.9496e+002 (2.8910e+002)	**6.6767e+001** **(2.4387e+002)**
f_5	4.8127e+003 (4.4511e+001)	**2.1846e-002** **(1.6000e-002)**
f_6	**6.2365e+001** **(1.4530e+001)**	7.0721e+001 (1.5324e+001)
f_7	**2.9061e+001** **(1.2393e+000)**	3.8296e+001 (1.0916e+000)
f_8	9.1062e+005 (1.2477e+005)	**8.7510e+005** **(1.3349e+005)**
f_9	1.0825e+001 (1.1886e+000)	**5.5308e+000** **(1.3665e+000)**
f_{10}	4.5201e+002 (4.9538e+001)	**3.4388e+002** **(3.1475e+001)**
f_{11}	9.2079e+002 (2.2923e+001)	**8.2431e+002** **(3.3237e+000)**
f_{12}	9.1608e+002 (1.3797e+001)	**8.2650e+002** **(5.9936e+000)**
f_{13}	9.1667e+002 (9.0488e+000)	**8.2615e+002** **(3.4888e+000)**
f_{14}	9.1363e+002 (1.6512e+001)	**5.3446e+002** **(8.6952e+001)**
f_{15}	9.9539e+002 (9.5791e+000)	**2.1832e+002** **(5.3872e+000)**

5 CONCLUSION

In this paper, a new variant of PSO is proposed for solving numerical optimizations. The two sub-swarms will increase population's diversity for prevent particles from fall into local optimum. In the experimental results, fifteen test functions of CEC 2005 are selected. In order to test the performance of proposed method, the SPSO 2011 is taken into comparison. From the results, it can be observed that the proposed method is able to explore better solutions.

ACKNOWLEDGEMENT

This work was supported in part by National Science Council of Taiwan, Taiwan, R.O.C. under Grant NSC 101-2221-E-161-011.

REFERENCES

[1] J. H. Holland , Adaptation in Natural and Artificial Systems , University of Michigan Press, 1975

[2] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc.* of the 1st European Conference on Artificial Life, pp. 134-142, 1991.

[3] J. Kennedy and R. Eberhar, "Particle Swarm Optimization," in *Proc.* of *IEEE Int. Conf. on Neural Network*, pp. 1942-1948, 1995.

[4] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[5] D. Karaboga, "An idea based on honeybee swarm for numerical optimization," *Technical Report* TR06, 2005.

[6] M. G. H. Omran , A. P. Engelbrecht and A.Salman, " Differential Evolution Based Particle Swarm Optimization," in *Proc.* of *IEEE Swarm Intelligence Symposium*, pp.112-119, 2007.

[7] C. H. Chen and S. N. Yeh "Personal Best Oriented Constriction Type Particle Swarm Optimization," in *Proc.* of *IEEE Conference on Cybernetics and Intelligent Systems*, pp.1-4, 2006.

[8] D. Bratton and J. Kennedy "Defining a Standard for Particle Swarm Optimization," in *Proc.* of *IEEE Swarm Intelligence Symposium*, pp.120-127, 2007.

[9] P. Ghosh, H. Zafar, S. Das and A. Abraham, "Hierarchical dynamic neighborhood based Particle Swarm Optimization for global optimization," in *Proc.* of *IEEE Congress on Evolutionary Computation (CEC)*, pp.757-764, 2011.

[10] T. M. Blackwell and J. Branke, "Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments," in *Proc.* of *IEEE Transactions on Evolutionary Computation*, vol. 10/4, pp. 51-58, 2004.

[11] Y. Shi, and R. Eberhart, " A modified particle swarm optimizer", in *Proc.* of *Conference on Evolutionary Computation: IEEE Press. Piscataway.* , pp.69 –73, 1998.

[12] [Online]. Available: http://particleswarm.info

[13] [Online]. http://www.ntu.edu.sg/home/EPNSugan