Particle Swarm Optimization with adjusting Scope of Particles Distribution Automatically

Chiabwoot. Ratanavilisagul, Poonsak. Wongsawasdi, Wirote. Lerttheerachanchai, Preeyaporn. Mutchika and Narin. Jeeranantasin

Abstract— Particle swarm optimization (PSO) is utilized to solve optimization problems such as Knapsack Problem and Travelling Salesman Problem. PSO can solve optimization problems effectively. However, it encounters the problem of trapping in local optimum. In order to overcome this problem, particles reposition technique applies with PSO (PSORPG). Nevertheless, PSORPG suffers complexity of adjusting scope of particles distribution after reposition is executed. This research proposes improvement PSORPG. This proposed technique, the scope of particles distribution can be adjusted to solve the problem appropriately and it is called PSORPGA. The proposed technique is tested on twenty-six benchmark functions and gets gratified search results.

Keywords—Swarm Intelligence, Particle Swarm Optimization, Mutation Operator, Particles Re-initialization, Multi-start Particles, Particles Reposition.

I. Introduction

PARTICLE Swarm Optimization is proposed by Kennedy and Eberhart in 1995 [1], [2]. It is motivated from foraging of the bird's flocks. In present, it is interested from many researchers and applied with many fields [3] such as Knapsack Problem [6] and Travelling Salesman Problem [12]**.** It is arranged in group of population-based stochastic optimization and evolutionary algorithm. When it is compared with several techniques in same group [4], such as genetic algorithm (GA), PSO performs better in solving many optimization problems with fast and stable convergence rate.

PSO has many advantages such as rapid convergence, simplicity, and little parameters to be adjusted [5], [6]. It has disadvantage that is trapping in local optimum or premature convergence when it is used to solve complex multimodal problems [5], [6]. To overcome drawbacks of PSO, many researchers [5], [6], [7], [8], [9] increase searching diversity in the population of PSO by adding the particles re-initialization technique (re-initialization) or particles reposition technique (reposition) into the process of PSO. The experimental results of these researches showed that these techniques can increase performance searching of PSO and get the better solutions. Both reposition and re-initialization can be called that resetting.

However, resetting encounters parameters adjusting to suitably deal with any given problem. The essential parameter is the resetting probability. This parameter affect with scope of particles distribution. If this parameter is over assigned, particles distribute in wide areas. It wastes much time for convergence of particles. On the other hand, if this parameter is under assigned, particles distribute in narrow areas. It cannot jump out local optimum, so adding this technique in the process of PSO is useless.

Researchers proposed various ways to define the resetting probability, such as constant dimension [8] and constant probability [6]. The weak point of algorithm’s [6], the resetting probability is constant value forever running, so it cannot automatically adjust this value to appropriate with solved problem. This paper proposes adjusting the resetting probability automatically following the result from resetting current round and previous round to suitably deal with any given problem. A set of benchmark test functions is used to compare the standard PSO [1], [2], PSO algorithm’s [8], and PSO algorithm’s [6], and the proposed PSO algorithm. The results show that the proposed PSO algorithm obtains the best results in all test functions.

The rest of this paper is organized as follows. Section 2 explains the standard PSO and PSO with resetting. Section 3 explains PSO with adjusting scope of particles distribution automatically. Section 4 explains the experiment setup and presents the experiment results. Section 5 concludes the paper with a brief summary.

Chiabwoot. Ratanavilisagul is with the Technology Computer Department, Faculty of Technical Education, Rajamangala University of Technology Krungthep (RMUTK), Bangkok, Thailand (e-mail: chaibwoot@hotmail.com).

Poonsak. Wongsawasdi is with the Technology Computer Department, Faculty of Technical Education, Rajamangala University of Technology Krungthep (RMUTK), Bangkok, Thailand (e-mail: pskw@rmutk.ac.th).

Wirote. Lerttheerachanchai is with the Technology Computer Department, Faculty of Technical Education, Rajamangala University of Technology Krungthep (RMUTK), Bangkok, Thailand (e-mail: wirotelert@rmutk.ac.th).

Preeyaporn. Mutchika is with the Technology Computer Department, Faculty of Technical Education, Rajamangala University of Technology Krungthep (RMUTK), Bangkok, Thailand (e-mail: preeyaporn.m@rmutk.ac.th).

Narin. Jeeranantasin is with the Technology Computer Department, Faculty of Technical Education, Rajamangala University of Technology Krungthep (RMUTK), Bangkok, Thailand (e-mail: narin_je@hotmail.com).

II. Related Work

A. Particle Swarm Optimization

In standard PSO, each member of the population is called a “particle” with its own position and velocity. Each individual particle performs searching in the search space according to its velocity, the best position found in the whole swarm (GBEST) and the individual's best position (PBEST). PSO starts with randomizing particle positions and their respective velocities, and the evaluation of the position of each particle is achieved using the objective function of the optimization problem. In a given iteration, each individual particle updates its position and velocity according to the expression below:

$$V_{id}{}' = \varpi V_{id} + \eta_1 rand()(P_{id} - X_{id}) + \eta_2 rand()(P_{gd} - X_{id}) \quad (1)$$

$$X_{id}{}' = X_{id} + V_{id}{}' \quad (2)$$

Where X'_{id} is the current positions of i particle and d dimension, X_{id} is the previous positions of i particle and d dimension, V'_{id} is the current velocity of i particle and d dimension, V_{id} is the previous velocity of i particle and d dimension, P_{id} is PBEST of i particle and d dimension, P_{gd} is GBEST of d dimension, $0 \le \omega < 1$ is an inertia weight, η_1 and η_2 are acceleration constants, and rand() generates random number from interval [0,1]. A limit velocity is represented with V_{max}.

B. Particle Swarm Optimization with re-initialization

Reference [8] proposed PSO with chaotic opposition-based population initialization and stochastic search technique (PSOCS). The concept of this algorithm can be summarized as follows: after update position process, all particles are reset. Amount of dimensions are reset. It is constant value forever searching. The position from resetting originates from randomness position on GBEST and PBEST that is not its PBEST. The position from resetting is compared with its PEBST. If this position is better than PEBST, PBEST is replaced by this position. This algorithm obtains better the quality of solutions. However, it cannot search for new area because the position from resetting originates from PEBST and GBEST that are old positions. Some problems, it may not succeed in finding the best solution.

C. Particle Swarm Optimization with reposition

The trapping in local optimum is possible for standard PSO. When PSO trapped in local optimum, GBEST is not updated at the point of trapping and is repeatedly produced forever searching time. Normally, resetting should be used when swarm is trapped in order to distribute particles to search for solutions in new areas [5], [6].

In case of re-initialization technique requires resetting particles or velocity or GBEST or PBEST, so it needs to use a significant amount of time for the particles to converge in each reset [6]. Reference [6] proposed apply particles reposition technique with standard PSO (PSORPG) to quicken the convergence in each resetting. The experimental results showed that reposition gets solutions better than re-initialization.

PSORPG has two additional parameters from standard PSO. The first parameter is the threshold of restart (TR). Assignment TR is easy because TR is assigned much value to execute reposition when PSO traps in local optimum. The second parameter is the probability of reposition (PR). Assignment PR does not have obvious criterion, so adjusting PR with general problems is complicated.

III. Particle Swarm Optimization with adjusting Scope of Particles Distribution Automatically

As previously mentioned, adjusting PR with general problems is complicated. In addition, PR is constant value forever running, so PR cannot adjust value appropriately with solving problems. If this value is wrong defined, reposition technique is useless or particles waste much time for convergence. Hence, PR should be automatically adjusted to solve problems appropriately.

Normally, if PSORPG does not encounter trapping in local optimum, GBEST is improved until it can meet the best solution. But, if PSORPG encounter trapping in local optimum, GBEST is stagnant. In next time, PSORPG will execute reposition to solve trapping in local optimum. So, GBEST can identify particles distribution boundary that is enough for solving.

If boundary is enough, GBEST of current reset round is better than GBEST of pervious reset round because particles distribution is enough to meet new solution. In this case, boundary or PR should be decrease to reduce time for convergence.

If boundary is not enough, GBEST of current reset round is worse or equal than GBEST of pervious reset round because particles distribution is not enough to meet new solution. In this case, boundary or PR should be increase to enhance sufficient boundary to search for new solution.

Thereby, this paper proposed a novel dynamic PR to increase performance in searching of PSORPG. Moreover, this technique can decrease complication of adjusting PR. The proposed technique is called that PSORPG with adjusting probability of reposition automatically (PSORPGAP).

The concept of proposed technique is explained as follows: the begin phase, PR set to be small amount in order that convergence time is little. If PSORPG cannot search for global optimum, PSORPG is trap in local optimum then PSORPG executes reposition. After executed reposition, if GBEST of current reset round is more than GBEST of pervious reset round, PR diminishes. On the other hand, if GBEST of current reset round is less than or equal GBEST of pervious reset round, PR enhance. For Pseudo code of PSORPGAP is shown below:

```
Initial particles of each particle
While (termination condition ≠ true) do
    Evaluate the fitness of each particle
    If fitness of each particle is better than PBEST, update PBEST
    If fitness of each particle is better than GBEST, update GBEST
    Update each particle position according to (1) and (2)
    If times of GBEST consecutive unchanged ≥ TR
        If Fitness of PGBEST >= Fitness of GBEST
            PR = PR + PPR;
        Else
```

```
            PR = PR - PPR;
        End if
        If PR > MAXPR
            PR = MAXPR
        End if
        If PR < MINPR
            PR = MINPR
          End if
        PGBEST = GBEST;
          Reset GBEST, PBEST
        For i = 1 to N
            For d =1 to D
            If PR > rand() then
                Apply (3)
            End if
          Next d
        Next i
      End if
    End while
```

Where N is size of the population, D is dimension of the solution space. TR is the threshold of reposition. PR is the probability of reposition. PGBEST is the GBEST of the previous reset round. PPR is the probability of reposition for increasing and decreasing in each reset round. MAXPR is the maximum value of PR. MINPR is the minimum value of PR.

$$X_{id} = X_{id} \pm (X_{id} \times rand()) \qquad (3)$$

Equation (3) is reposition equations in which the positive operator is selected if the random number generated uniformly in the range [0, 1] is less than 0.5 and the negative operator otherwise.

IV. Experiments and Results

The proposed algorithm is tested on twenty-six well-known benchmark functions [10], [11] listed in Table 1. These functions consist of seventeen multimodal functions from function one to seventeen. The remaining nine functions are unimodal functions. The search spaces, objective function value, number of dimension and V_{max} of this experiment are defined in Table 1.

Parameters are as follows for all experiments: η_1 and η_2 are both set to be 1.496180 and ω = 0.729844. Population size set to be 60. The number of experiments of each function is 100 runs. The number of evaluations is set to 200000 for dimensions are more than or equal 10 dimensions. The remaining is set to 10000.

The non-PSO parameters are as follows: parameters of PSOCS are set according to suggested by the original papers [8]. Parameters of PSORPG are set according to suggested by the original papers [6] except PR set to be 30. Parameters of PSORPGAP, MAXPR = 0.95, MINPR = 0.05, PPR = 0.05, PR = 0.05, TR = 100.

This research uses a personal computer with Intel Core I7 3770 with a 2.4-GHz CPU and 8 GB RAM, and Visual C++ 2010 as the programming language.

The measures of algorithm performance in the experiments are as follows:

The mean best fitness value (MBF) is the mean of best fitness in the final iteration from all running and is indicative of solution searching efficiency of an algorithm. In the case of experiments with the benchmark functions, all functions have a zero as minimum point. In Table 1, an entry less than 10^{-50} is given the value of zero. Meanwhile, the closer the MBF to the zero point of a method, the better the method.

TABLE I
Details of benchmark test functions

Function Name	Dim	Search space [X_{max}, X_{min}]	Objective function value	Attribute	V_{max}
ACKLEY	40	$x \in [-32.768, 32.768]^n$	0	Multimodal	32.768
GRIEWANK	40	$x \in [-300, 300]^n$	0	Multimodal	300
RASTRIGIN	40	$x \in [-5.12, 5.12]^n$	0	Multimodal	5.12
ROSENBROCK	40	$x \in [-2.048, 2.048]^n$	0	Multimodal	2.048
SCHWEFEL	40	$x \in [-500, 500]^n$	0	Multimodal	500
COSINE MIXTURE	40	$x \in [-1, 1]^n$	0	Multimodal	1
EXPONENTIAL	40	$x \in [-1, 1]^n$	0	Multimodal	1
LEVY	40	$x \in [-10, 10]^n$	0	Multimodal	10
DIXON-PRICE	40	$x \in [-10, 10]^n$	0	Multimodal	10
MICHALEWICZ	10	$x \in [-0, \lambda]^n$	0	Multimodal	λ
STEP	100	$x \in [-5.12, 5.12]^n$	0	Multimodal	5.12
SCHAFFER	2	$x \in [-100, 100]^n$	0	Multimodal	100
HOLDER	2	$x \in [-10, 10]^n$	0	Multimodal	10
BEALE	2	$x \in [-4.5, 4.5]^n$	0	Multimodal	4.5
SHUBERT	2	$x \in [-10, 10]^n$	0	Multimodal	10
GOLDSTEIN-PRICE	2	$x \in [-2, 2]^n$	0	Multimodal	2
SIX-HUMP CAMEL	2	$x \in [-2, 2]^n$	0	Multimodal	2
SPHERE	40	$x \in [-5.12, 5.12]^n$	0	Unimodal	5.12
PARALLEL HYPER-ELLIPSOID	40	$x \in [-5.12, 5.12]^n$	0	Unimodal	5.12
ROTATED HYPER-ELLIPSOID	40	$x \in [-65.536, 65.536]^n$	0	Unimodal	65.536
CIGAR	40	$x \in [-10, 10]^n$	0	Unimodal	10
BROWN	40	$x \in [-1, 4]^n$	0	Unimodal	4
MULTIMOD	40	$x \in [-10, 10]^n$	0	Unimodal	10
ZAKHAROV	40	$x \in [-5, 10]^n$	0	Unimodal	10
TRID	10	$x \in [-100, 100]^n$	0	Unimodal	100
EASOM	2	$x \in [-100, 100]^n$	0	Unimodal	100

The mean the probability of reposition (MPR) is the mean of the probability of reposition in the final iteration from all running.

TABLE II
COMPARATIVE RESULTS OF PSORPGAP, PSO, PSOCS AND PSORPG

TECHNIQUES	PSO	PSOCS	PSORPG	PSORPGAP	
Function Name	MBF	MBF	MBF	MBF	MPR
ACKLEY	1.14003	0.001003	6.68E-15	3.48E-15	0.444
GRIEWANK	0.019515	0.003100	0.00698	1.85E-05	0.635
RASTRIGIN	130.61	4.46704	9.69708	3.62877	0.313
ROSENBROCK	1.25E-11	0.025155	5.72E-12	1.56E-12	0.119
SCHWEFEL	3069.48	2412.072	2452.35	2356.07	0.25
COSINE MIXTURE	0.872285	1.52E-09	8.88E-18	0	0.537
EXPONENTIAL	2.42E-16	6.29E-10	2.22E-18	0	0.512
LEVY	19.1263	0.048216	0.019814	1.29E-06	0.422
DIXON-PRICE	0	0	0	0	0.05
MICHALEWICZ	1.19E-09	0	0	0	0.077
STEP	861.66	48.62	0	0	0.183
SCHAFFER	0.0031	0.0081	0.003373	0.003083	0.05
HOLDER	0	0	0	0	0.05
BEALE	3.64E-16	4.52E-06	1.55E-12	1.82E-16	0.05
SHUBERT	0	0.008468	0	0	0.05
GOLDSTEIN-PRICE	0	5.95E-06	0	0	0.05
SIX-HUMP CAMEL	0	0	0	0	0.05
SPHERE	0	3.34E-08	0	0	0.052
PARALLEL HYPER-ELLIPSOID	0	5.17E-07	0	0	0.052
ROTATED HYPER-ELLIPSOID	1.4E-45	5.10E-05	0	0	0.05
CIGAR	5.61E-41	0.009063	0	0	0.053
BROWN	1.4E-45	8.79E-09	0	0	0.052
MULTIMOD	0.001047	15.477	1.20E-14	1.46E-21	0.056
ZAKHAROV	0.000106	92.1654	5.33E-05	2.87E-05	0.05
TRID	0.002949	1.18083	0.00041	0.00012	0.417
EASOM	0.000852	0.269973	0.04	0.02	0.05

From the experimental results in Table 2, the results of MBF show that PSORPGAP outperforms PSO, PSOCS, and PSORPG for both multimodal functions and unimodel functions because its lower MBF than them.

For all unimodel functions and some multimodal functions, PSOCS is poorer MBF than PSO such as ROSENBROCK, SCHAFFER, BEALE, SHUBERT, and GOLDSTEIN-PRICE because PSOCS loses evaluation calls for resetting in each particle. PSOCS is less convergence speed than PSO.

For GRIEWANK, COSINE MIXTURE, EXPONENTIAL, and LEVY, PSORPGAP is lower MBF than PSORPG. PR of PSORPG set to be 0.3. The results from MPR of PSORPGAP show that PR should have value more than 0.4. Hence, PR of PSORPG is defined too little for solving these problems. Particles may be not able to jump out local optimum. The reposition technique is inefficient.

For BEALE and MULTIMOD, PSORPGAP is lower MBF than PSORPG. PR of PSORPG set to be 0.3. The results from MPR of PSORPGAP show that PR should have value about 0.05 or less than. Hence, PR of PSORPG is defined too much for solving these problems. It wastes much time for convergence of particles after resetting.

Both multimodal functions and unimodel functions, PSORPGAP outperforms PSORPG. PSORPGAP is different from PSORPG, it can automatically adjust PR. So, from this experiment show that PR affects to search for solutions. In addition, PSORPGAP can adjust PR to appropriate with solved problems.

V. CONCLUSION

The reposition technique applied with PSO to solve problems of trapping in local optimum. However, applying the particles reposition with general problems is complicated that is the cause of complicatedly adjusting PR. It affects to search for solution. This paper proposes PR is automatically adjusted according to result from search for solution in each reset round. The proposed technique is called PSORPGAP. From the experimental results show that PR affects to search for solution and PSORPGAP outperforms PSORPG, PSOCS, and PSO.

REFERENCES

[1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[2] J. Kennedy and R. C. Eberhart, "A New Optimizer Using Particle Swarm Theory," *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.

[3] R. Poli, J. Kennedy, T. Blackwell, "Particle swarm optimization," *Swarm Intelligent*, pp. 33–57, 2007.

[4] R.C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *The 7th Annual Conference on Evolutionary Programming*, 1998.

[5] R. Chiabwoot and K. Boontee, "A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period," *Advances in Intelligent Systems and Computing*, pp. 33-42, 2014.

[6] R. Chiabwoot and K. Boontee, "A modified particle swarm optimization with mutation and reposition," *International Journal of Innovative Computing Information and Contro*l, pp. 1–18, 2014.

[7] J. Guo, S. j. Tang, "An Improved Particle Swarm Optimization with Re-initialization Mechanism," *Intelligent Human-Machine Systems and Cybernetics*, pp. 437–441, 2009.

[8] W. Gao, S. Liu and L. Huang, "Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique," *Communications in Nonlinear Science and Numerical Simulation*, pp. 4316–4327, 2012.

[9] K. K. Budhraja et al., "Exploration Enhanced Particle Swarm Optimization using Guided Re-Initialization," *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications, Advances in Intelligent Systems and Computing*, pp. 403–416, 2013.

[10] Test Functions. http://www-optima.amp.i.kyoto-u.ac.jp/member/student /hedar/Hedar_files/TestGO_files/ Page364.htm

[11] Test Functions and Datasets. http://www.sfu.ca/~ssurjano/index.html

[12] S. Jaturong and T. Panwit, "An improvement Particle Swarm Optimization for Travelling Salesman Problem with the mutation operator," *International Conference on Emerging Trends in Computer and Image Processing (ICETCIP'2014)*, pp. 12–14, 2014.