A Modified Particle Swarm Optimization with Dynamic Mutation Period

C. Ratanavilisagul and B. Kruatrachue, "A modified particle swarm optimization with dynamic mutation period," 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Nakhon Ratchasima, Thailand, 2014, pp. 1-6, doi: 10.1109/ECTICon.2014.6839762.

Chiabwoot Ratanavilisagul
Dept. of Computer Engineering
Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand
chaibwoot@hotmail.com

Boontee Kruatrachue
Dept. of Computer Engineering
Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand
booontee@yahoo.com

***Abstract*— the particle swarm optimization (PSO) is an algorithm that attempts to search for better solution in the solution space by attracting particles to converge toward a particle with the best fitness. PSO is typically troubled with the problems of trapping in local optimum and premature convergence. In order to overcome both problems, we propose an improved PSO algorithm that is applied mutation operator dynamically when particles are in local optimum. Moreover, the mutation period can be adjusted to solve the problem appropriately. The proposed technique is tested on benchmark functions and gives more satisfied search results in comparison with PSOs for the benchmark functions.**

Keywords*— *Swarm Intelligence; Particle Swarm Optimization; Cauchy Mutation; Mutation Operator.

I. INTRODUCTION

Kennedy and Eberhart [1], [2] introduced Particle Swarm Optimization (PSO) in 1995. It is motivated by the behavior of flying bird and their communication mechanism in solving optimization problems. In comparison with several other population-based stochastic optimization methods, such as genetic algorithms (GAs) and evolutionary programming (EP), PSO performs better in solving a variety of optimization problems with rapid and stable convergence rate [3]-[5]. In addition, PSO is capable of solving the function optimization problems [6]-[8], neural network training [9], [10], as well as pattern classification and fuzzy system control [11], [12].

PSO has many advantages such as rapid convergence, simplicity, and little parameters to be adjusted. Its main disadvantage is having chances to converge on a local optimum, which leads to premature convergence. To overcome this disadvantage, many researchers [14]-[21] increased searching diversity in the population of PSO by adding the mutation operators in the process of PSO. The result showed that the proposed method can increase PSO optimization performance.

However, the mutation operators encounter parameters adjusting to suitably deal with any given problem since inappropriate parameters could contribute to poorer results than the standard PSO. The essential parameter of mutation operators is the mutation period.

Researchers proposed various ways to define the mutation period based on constant mutation rate [17], the best particle that is stagnant [17], and particle diversity [18].

This paper investigates adjusting mutation period of PSO to suitably deal with any given problem. Normally, the mutation should occur when swarm traps in local optimum. Hence, this paper proposes the mutation period defined by the unchanged number of the best consecutive particle which indicates the state of local optimum trapping. Moreover, mutation period is automatically adjusted according to the difference of the best solution found after mutation and the previous best solution.

A set of benchmark test functions is used to compare the standard PSO, the proposed PSO algorithm, PSO with mutation based on constant mutation rate (PSOMC) [17], PSO with mutation based on the best particle that is stagnant (PSOMS) [17], and PSO with mutation based on particle diversity (PSOMD) [18]. 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 basic PSO, the mutation operator, and elements of mutation operator. Section 3 explains PSO with dynamic mutation period (the proposed PSO algorithm). Section 4 explains the benchmark functions, the experiment setup and presents the experiment results. Section 5 concludes the paper with a brief summary.

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). The standard PSO algorithm starts with randomizing particle positions and their respective velocities, and the evaluation of the position of each particle is achieved by 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} represents the current positions of i particle and d dimension, X_{id} represents the previous positions of i particle and d dimension, V_{id} represents the previous velocity of i particle and d dimension, V'_{id} represents the current velocity of i particle and d dimension, P_{id} represents PBEST of i particle and d dimension, and P_{gd} represents GBEST of d dimension. η_1 and η_2 are acceleration constants, $0 \leq \varpi < 1$ is an inertia weight and rand() generates random number from interval [0,1]. A limit velocity is represented with V_{min}, V_{max}. If calculate velocity of a particle exceeds this value, it will replace value of V_{min}, V_{max}.

B. *Mutation Operators*

The mutation operator is the concept of genetic algorithm (GA). The principle of mutation operator is to randomly change positions of particles a little in some dimension within population. The lack of population diversity in PSO algorithms is a factor in their convergence on local optimum [13]. The mutation operator is added in process of PSO in order to increase population diversity. Therefore, it expands the search space and prevents convergence to local optimum [22]. It also increases the chance to escape from local optimum, so it improves the effectiveness of PSO [23].

C. *Elements of Mutation Operators*

The mutation applied to PSO has elements of mutation as follows:

1) The mutation equation is the equation designed for computing mutation to apply to some dimension of position or velocity in PSO. Many researchers proposed various mutation equations such as: N. Higashi and H. Iba. [20] implement a mutation equation from Gaussian distribution, so it is called Gaussian mutation. A. Stacey, M. Jancic, and I. Grundy. [21] implement a mutation equation from Cauchy distribution, so it is called Cauchy mutation. The results from different mutation equation affect the problem solving differently such as Cauchy mutation is suitable for multimodal problems while Power mutation is suitable for unimodal problems [16]. In order to solve multimodal problems efficiently, this paper selects Cauchy mutation.

2) The mutation point is position (particle positions, PBEST, GBEST) or velocity which can be mutated. Normally, the mutation is applied to particle positions.

3) The mutation amount is a number of dimensions of particles in population which is mutated in each mutation period. Many researchers proposed various mutation amounts such as: MO Lin. and ZHENG Hua. [18] used P_m to determine the mutation amount. Each dimension of particles in population will be mutated when random value on interval [0, 1] is less than P_m. P_m is taken as random value on interval [0.1, 0.3]. Andrews P S. [17] used mutation probability to determine the mutation amount. Each dimension of particles in population will be mutated when random value on interval [0, 1] is less than mutation probability. The mutation probability is defined by using the following (3). Normally, the mutation amount is float point, so adjusting this parameter to suitably deal with any given problem is complicated.

$$M_p = \frac{M_r}{M \times D} \quad (3)$$

Where M represents the size of the population, D represents the dimension of the solution space, M_p represents the mutation probability, and M_r represents the mutation rate.

4) The mutation period is condition to decide which iteration is executed mutation or not. If the setting of this parameter causes over-mutation, this obstructs the convergence of particles and thus induces a random search, the result of which is poorer than that of standard PSO. On the other hand, under-mutation, provide too little mutation to escape from trapping in local optimum, thereby mutation is useless. Many researchers suggested setting of the mutation period in order to improve performance of the searching of PSO such as:

Andrews P S. [17] proposed PSO with mutation based on constant mutation rate (PSOMC). The concept of this algorithm can be summarized as follows: The mutation is applied in every iteration after all the particle positions have been updated. For each particle and for each dimension of a particle conditionally mutate the particle position in this dimension equal to mutation probability according to (3). The mutation probability depends on the number of particles and dimensions per particle.

The object of PSOMC is to enhance the population diversity by applying mutation in every iteration. Since mutation is applied in every iteration, this can be over-mutation. Some problems may not need to apply mutation or apply mutation in some iteration. Hence, the results of this algorithm may be poorer than those of standard PSO.

Andrews P S. [17] proposed PSO with mutation based on the best particle that is stagnant (PSOMS) [17]. The concept of this algorithm can be summarized as follows: Applying mutation is similar to PSOMC except the mutation is applied when the number of iteration that GBEST remains unchanged, is greater than some threshold value.

The object of PSOMS is to create particles jump out local minima when swarm traps in local optimum. This algorithm could adjust mutation period suitably for solving problems. The main problem of this algorithm is to identify threshold value suitable for a problem. If threshold value is inappropriately defined, the results of this algorithm may be poorer than those of standard PSO.

MO Lin. and ZHENG Hua. [18] proposed PSO with mutation based on particle diversity (PSOMD). The concept of this algorithm can be summarized as follows: The mutation is applied in each iteration after all the particle positions have been updated if the population average distance amongst points (D(t)) [24] is less than some threshold (D_{min}). For each particle and for each dimension of particle, if a random number uniformly generated in the range [0:1] is less than P_m (P_m is random value on interval [0.1, 0.3]), then the particle position in this dimension is mutated.

The object of PSOMD is to preserve the population diversity. This algorithm enhances the population diversity when it detects population diversity less than threshold value (D_{min}). The main problem of this algorithm is to identify threshold value suitable for a problem. If threshold value is inappropriately defined, the results of this algorithm may be poorer than those of standard PSO. The population average distance amongst points (D(t)) is described as following:

$$D(t) = \frac{1}{M \times L} \sum_{i=1}^{M} \sqrt{\sum_{d=1}^{D} (x_{id}^{t} - \overline{p}_{id})^2} \tag{4}$$

Where L represents the diagonal length of search space, M represents the size of the population, D represents the dimension of the solution space, x^t_{id} represents the d dimension coordinate values of the i particle, and $\overline{p}_{id}$ represents the average value of the d dimension coordinate values of all particles.

III. PSO WITH DYNAMIC MUTATION PERIOD

As previously mentioned, the main problem of applying mutation operator technique is parameters adjusting. To avoid this problem, the proposed algorithm automatically adjusts its parameters.

The trapping in local optimum is possible for standard PSO, which leads to the stagnation of the searching in which the solution (GBEST) obtained at the point of trapping, is repeatedly produced irrespective of the length of searching time. Therefore, the unchanged number of consecutive GBEST can indicate state of swarm when swarm traps in local optimum.

Normally, the mutation operator should be used when swarm trap or trend to trap in local optimum in order to distribute particles to search in other areas. The easy techniques to indicate the trapping state of the swarm is to monitor the unchanged number of consecutive GBEST. Thereby, this paper proposed the mutation period considered from the unchanged number of consecutive GBEST.

If mutation period was set too small, particles will be reset before convergence. On the other hand, if mutation period was set too large, particles remain trapping in local optimum instead of search in other areas.

Therefore, this paper proposes the novel technique which can decrease problem of adjusting mutation period. The proposed technique can adjust the mutation period to solve any optimization problems appropriately.

The concept of proposed technique is explained as follows: before mutation, if the best solution in current mutation round is worse than previous round GBEST, particles could not converge to the GBEST. Hence, the mutation period should be increased to enhance convergence time of particles to the GBEST in next mutation round. On the other hand, if the best solution in current mutation round equals GBEST, particles can converge to the GBEST. In this case, the mutation period should be decreased to reduce the searching time in local optimum. The proposed technique is called PSO with dynamic mutation period (PSODMP). Pseudo code of PSODMP is shown below:

```
Initial position and velocity of each particle
Initial PBEST, GBEST, RBEST
While termination condition ≠ true do
  For each particle
    Evaluate particle fitness
    If fitness of each particle is better than that of PBEST
      Update PBEST (Pi=Xi)
    End If
    If fitness of each particle is better than that of GBEST
      Update GBEST (Pg=Xi)
    End If
    If fitness of each particle is better than RBEST
      Update RBEST (RBEST=Xi)
      TRBEST = -1
    End If
    Update particle position according to formula (1) and (2)
  End For
  TRBEST ++
  If TRBEST >= MP
    If fitness of RBEST is worse than fitness of GBEST
      MP++
    Else
      MP--
    End If
    If MAXMP < MP
      MP = MAXMP
    End If
    If MINMP > MP
      MP = MINMP
    End If
    Reset RBEST
    For each particle
      If percent mutation < rand() then
        Apply mutation according to formula (5)
      End If
    End For
    TRBEST = 0
  End If
End while
```

Equation (5) is mutation equations used in PSODMP 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

$$mutation\ (X_{id}) = X_{id} \pm Cauchy\ (\delta) \tag{5}$$

Where δ is a scale parameter that determines the shape of the distribution. Cauchy (δ) denotes Cauchy random numbers with the scale parameter of 1. The upper and lower limit positions are X_{max}, X_{min}. MP is the mutation period. RBEST is the best solution in each mutation round. TRBEST is the times of RBEST consecutive unchanged. MINMP is the minimum value of the mutation period. MAXMP is the maximum value of the mutation period.

IV. Experiments and Results

A. Benchmark Functions

The proposed algorithm is tested on ten well-known benchmark functions, listed in Table 1. These functions consist of seven multimodal functions from function one to seven. The remaining three functions are unimodal functions. The results of PSODMP on the benchmark functions are compared with PSO, PSOMC, PSOMS, and PSOMD.

TABLE I. DETAILS OF BENCHMARK TEST FUNCTIONS.

Problem no.	Function name	Expression
1	Ackley	$f(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$
2	Griewank	$f(x) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos(x_i/\sqrt{i}) + 1$
3	Rastrigrin	$f(x) = 10n + \sum_{i=1}^{n}(x^2 - 10\cos(2\pi x_i))$
4	Rosenbrock	$f(x) = \sum_{i=1}^{n}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
5	Schwefel	$f(x) = 418.9829 \times n + \sum_{i=1}^{n}(x_i \times \sin(\sqrt{\lvert x_i \rvert}))$
6	Step	$f(x) = \sum_{i=0}^{n-1}(\lfloor x_i \rfloor + 0.5)^2$
7	Cosine Mixture	$f(x) = -0.1 \times \sum_{i=0}^{n}\cos(5\pi x_i) + \sum_{i=0}^{n} x^2 + 0.1n$
8	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$
9	Parallel Ellipsoid	$f(x) = \sum_{i=1}^{n}(i \times x_i^2)$
10	Rotated Ellipsoid	$f(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i} x_j^2)$

TABLE I. (CONTINUED).

Problem no.	Search space $[X_{min}, X_{max}]$	Objective function value	Dim.	V_{min}, V_{max}
1	$x \in [-32.768, 32.768]^n$	0	30	-32.768, 32.768
2	$x \in [-300, 300]^n$	0	30	-300, 300
3	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
4	$x \in [-2.048, 2.048]^n$	0	30	-2.048, 2.048
5	$x \in [-500, 500]^n$	0	30	-500, 500
6	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
7	$x \in [-1, 1]^n$	0	30	-1, 1
8	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
9	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
10	$x \in [-65.536, 65.536]^n$	0	30	-65.536, 65.536

B. Parameters Setting

Parameters are as follows for all experiments: η_1 and η_2 are both set to be 1.496180 and ω = 0.729844, as suggested by F. van den Bergh [25], the number of population is 80. The number of experiments of each function is 100 runs. The maximum iteration is 10000. To guarantee fairness of performance measurement, the mutation equation (5) is applied to all methods.

The non-PSO parameters are as follows: For PSODMP algorithm, MINMP = 2, MAXMP = 100, MP = 50, the mutation probability = 0.01. The compared algorithm parameters are set according to suggested by the original papers. For PSOMC algorithm, the mutation rate = 0.5. For PSOMS algorithm, the mutation rate = 0.5, threshold value = 5. For PSOMD algorithm, D_{min} = 0.001.

This research is conducted by a personal computer of Intel Core i5 2450 with a 2.5-GHz CPU and 8 GB RAM, and Visual C++ as the programming language.

In Table 2, the mutation period of PSODMP is not adjusted dynamically, but it is fixed in order to compare to results from mutation period adjusting of PSODMP. To guarantee fairness of performance measurement, numbers of evaluation of all the algorithms are set equation. The number of evaluation of each run in experiments is 800000 evaluations.

TABLE II. THE RESULTS OF PSODMP BY FIXED MP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO	PSODMP	PSODMP	PSODMP	PSODMP	PSODMP
MP	-	2	5	10	50	100
Problem no.	MBF	MBF	MBF	MBF	MBF	MBF
1	0.330081	7.28E-15	7.57E-15	7.82E-15	8.67E-15	8.74E-15
2	0.0119302	0.00167542	0.00640026	0.00806867	0.0116821	0.0117401
3	73.2487	0.0320216	1.71E-15	7.39E-15	5.57E-14	7.16E-14
4	3.32E-30	0.011635	2.70E-19	1.34E-29	7.64E-31	3.45E-31
5	2008.25	37.1888	59.3283	170.483	693.962	940.314
6	6.00E-02	0	0	0	0	0
7	0.378328	0	0	0	3.55E-17	5.33E-17
8	1.46E-251	9.32E-219	1.20E-245	1.06E-251	8.98E-251	7.40E-252
9	1.57E-249	3.74E-217	8.34E-248	1.64E-249	2.24E-249	2.65E-249
10	2.70E-255	6.03E-223	3.67E-253	2.02E-255	1.34E-257	9.30E-256

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 (100 runs). MBF indicates the solution searching efficiency of an algorithm. All experiments benchmark functions have zero results as a minimum point. In Table 1, an entry less than 10^{-324} is given the value of zero. The closer the MBF to the zero point of a method, the better the method. The mean mutation period (MMP) is the mean of mutation period in every iteration from all running. MMP is the results of the calculation of PSODMP algorithm. SD is the standard deviation. SD indicates the solution searching reliability of an algorithm. MD(t) is the mean of the population average distance amongst points in the final iteration from all running. MD(t) indicates the diversity of algorithms. Student's T-Test (ST) is a statistical method that is used to see if two sets of data differ significantly.

From the experimental results of Table 2 show some functions, if the setting mutation period is 2 or 5, swarm will be over-mutation. Its searching solutions are poorer than results of the standard PSO, such as Rosenbrock function, Sphere function, Parallel Ellipsoid function, and Rotated Ellipsoid function. This is due to particles converge hardly. In the case of mutation period is equal or more than 50 in unimodal functions, overall its searching solutions are similar to results of the standard PSO since the mutation operator is not executed. For multimodal problems, if the setting mutation period is more than 50, swarm will be under-mutation. Its overall results are poorer than results of the mutation period setting ranged from 5 to 10, such as Ackley function, Griewank function, Rastrigrin function, Schwefel function, and Cosine function. Since swarm wastes time searching in local optimum rather than starting new search in other areas.

TABLE III. COMPARATIVE RESULTS OF PSO, PSOMC, PSOMS, PSOMD AND PSODMP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO	PSOMC	PSOMS	PSOMD	PSODMP	
Problem no.	MBF	MBF	MBF	MBF	MBF	MMP
1	0.330081	8.99E-15	8.99E-15	7.06E-09	7.53E-15	2.30462
2	0.0119302	0.0123409	0.00905385	0.00221653	0.00152787	2.59979
3	73.2487	1.36E-14	4.43E-14	9.58144	3.41E-15	6.56945
4	3.32E-30	2.96E-31	4.44E-31	0.000106823	1.48E-31	51.1689
5	2008.25	1424.73	1423.54	1303.92	7.46E+01	2.4443
6	6.00E-02	0	0	0.01	0	48.6433
7	0.378328	0	2.66E-17	1.97E-13	0	10.8324
8	1.46E-251	1.29E-246	2.51E-249	1.02E-15	4.63E-251	50
9	1.57E-249	3.51E-246	3.84E-248	9.85E-15	2.31E-249	50
10	2.70E-255	5.73E-247	1.46E-254	5.14E-231	2.52E-257	50

For multimodal problems in Table 2, PSODMP gets good results because of the mutation period setting ranged from 2 to 50. The PSODMP algorithm can adjust mutation period in this range and it can obtain the good solutions as shown in Table 3. Therefore, the PSODMP can dynamically adjust to suitably deal with the problem.

From the experimental results of Table 3 show that PSODMP outperforms PSO, PSOMC, PSOMS, and PSOMD in multimodal function. PSODMP produces a better quality solution because of its lowest MBF in multimodal function.

For multimodal function in Table 3, some MBF results of PSOMD are poorer than the standard PSO such as Rosenbrock function. If mutation period adjusting to deal with problems appropriately. The solutions of PSOMD can obtain better than that of the standard PSO. However, mutation period adjusting of PSOMD to deal with any given problem appropriately is complicated.

For unimodel function in Table 3, PSO and PSODMP have similar overall results. The MBF results of PSO and PSODMP are higher than that of PSOMC, PSOMS, and PSOMD because these algorithms could execute the mutation operator which will hinder the convergence of particles in unimodel function. In case of unimodel function shows PSOMC, PSOMS, and PSOMD cannot adjust mutation period to deal with any given problem appropriately. Hence, the results of these algorithms could be poorer than standard PSO.

TABLE IV. COMPARATIVE RESULTS OF SD OF PSO, PSOMC, PSOMS, PSOMD AND PSODMP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO	PSOMC	PSOMS	PSOMD	PSODMP
Problem no.	SD	SD	SD	SD	SD
1	0.583675	3.15E-15	3.19E-15	3.83E-09	2.42E-15
2	0.0128751	0.0163161	0.0105423	0.0056868	0.0053869
3	19.5677	3.53E-14	4.33E-14	14.7998	1.35E-14
4	1.58E-29	4.61E-30	5.48E-30	4.34E-05	3.50E-30
5	569.192	344.588	365.627	328.79	84.618
6	0.698212	0	0	0.908625	0
7	0.20742	0	1.39E-16	1.25E-13	0
8	0	0	0	8.72E-16	0
9	0	0	0	1.63E-14	0
10	0	0	0	0	0

TABLE V. COMPARATIVE RESULTS OF MD(T) OF PSO, PSOMC, PSOMS, PSOMD AND PSODMP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO	PSOMC	PSOMS	PSOMD	PSODMP
Problem no.	MD(T)	MD(T)	MD(T)	MD(T)	MD(T)
1	2.84E-24	1.61882E-14	1.69556E-16	0.0203683	5.83E-06
2	0.00386739	0.031068	0.055874	0.131367	0.330386
3	2.71834E-17	1.93217	0.294151	0.0277783	0.079165
4	3.39155E-31	0.0124834	0.005104	0.21838	0.009747
5	2.38275E-14	2.33763	2.47937	35.6269	591.101
6	0.0230098	0.126497	0.088937	0.0324001	0.076241
7	3.07902E-17	1.97259E-09	7.58E-09	0.00408327	4.1E-09
8	1.4003E-251	6.95328E-17	3.79E-251	0.00922085	2.6E-250
9	5.6331E-251	4.6627E-137	2.34E-248	0.0089146	2.1E-250
10	1.30005E-08	0.279225	0.000365	0.420949	3.48E-05

TABLE VI. COMPARATIVE RESULTS OF ST BETWEEN PSODMP AND THE OTHER ALGORITHMS (PSO, PSOMC, PSOMS, PSOMD) ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Compared algorithms	PSO	PSOMC	PSOMS	PSOMD
	ST	ST	ST	ST
PSODMP	0.162301	0.17171	0.17171	0.169745

From the experimental results of Table 4 show that the reliability of PSODMP is better than that of PSO, PSOMC, PSOMS, and PSOMD because of its lowest SD in all test functions. Hence, PSODMP outperforms PSO, PSOMC, PSOMS, and PSOMD in case of reliability and quality solution.

For multimodal problems in Table 5, the population diversity of PSOMC, PSOMS, PSOMD and PSODMP are more than that of PSO because MD(t) of PSOMC, PSOMS, PSOMD and PSODMP in all multimodal functions are more than that of PSO. Hence, the mutation operator can improve the population diversity of PSO.

From the experimental results of Table 6 show that the statistically significant result of PSODMP is different from that of PSO, PSOMC, PSOMS, and PSOMD. Therefore, the ST results of PSO, PSOMC, PSOMS, and PSOMD have a little value when these algorithms are compared with PSODMP.

V. CONCLUSION

The mutation operator is applied with PSO to solve problems of trapping in local optimum and premature convergence. However, applying the mutation operator with general problems is complicated which is the cause of complicatedly adjusting the mutation period in order that the mutation operator can enhance the solution searching performance and can be easily applied to general problems. This paper proposes the mutation period that is automatically adjusted according to the difference of the best solution found after mutation and the previous best solution. The proposed technique is called PSODMP. On benchmark functions, the proposed PSODMP, PSO, PSOMC, PSOMS, and PSOMD are tested and the results are compared. From the experimental results show that PSODMP can adjust the mutation period to solve the problem appropriately and PSODMP outperforms PSO, PSOMC, PSOMS, and PSOMD.

REFERENCES

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

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

[3] K. E Parsopoulos, V. P. Plagianakos, G. D. Magoulas, M. N. Vrahatis, "Objective function stretching to alleviate convergence to local minima," Nonlinear Analysis TMA 47, 2003, pp. 3419-3424.

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

[5] J. Kennedy, R.C. Eberhart, "Swarm Intelligence, Morgan Kaufmann," San Mateo, 2001.

[6] V.S. Pappala, I. Erlich, K. Rohrig, J.A. Dobschinski, "A stochastic modal for the optimal operation of a wind–thermal power system," IEEE Trans. Power Syst., May 2009, pp. 940–950.

[7] P. Acharjee, S.K. Goswamj, "A decoupled power flow algorithm using particle swarm op-timization technique," Energy Convers. Manage, 2009, pp. 2351–2360.

[8] S. Dutta, S.P. Singh, "Optimal rescheduling of generators for congestion management based on particle swarm optimization," IEEE Trans. Power Syst., 2008.

[9] S. Kiranyaz, T. Ince, A. Yildirim, M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization," Neural Networks, 2009, pp. 1448–1462.

[10] H.L. Wei, S.A. Billings, Y.F. Zhao, L.Z. Guo, "Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio-temporal system identification," IEEE Trans. Neural Network, 2009, pp. 181–185.

[11] C.J. Lin, C.H. Chen, C.T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," IEEE Trans. Syst. Man Cybernetics Part C, 2009, pp. 55–68.

[12] L. Zhao, F. Qian, Y. Yang, Y. Zeng, H. Su, "Automatically extracting TS fuzzy models using cooperative random learning particle swarm optimization," Appl. Soft Comput., 2010, pp. 938–944.

[13] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," IEEE Transactions on Evolutionary Computation, vol. 8,no. 3, 2004, pp. 240–255.

[14] R. A. Krohling, "Gaussian particle swarm with jumps," in Proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2005, pp. 1226-1231.

[15] R. A. Krohling, L. dos Santos Coelho, "PSO-E: Particle Swarm with Exponential Distribution," Proceedings of the IEEE Congress on Evolutionary Computation, July 2006, pp. 1428- 1433.

[16] Muhammad Imran, Hajira Jabeen, Mubashir Ahmad, Qamar Abbas,Waqas Bangyal, Qamar Abbas, "Opposition based PSO and Mutation Operators," 2010 2nd International Conference on Education Technology and Computer (ICETC), pp. 506- 508.

[17] Andrews P S., "An investigation into mutation operators for particle swarm optimization," Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, 2006, pp. 1044-1051.

[18] MO Lin, ZHENG Hua., "Improved PSO Algorithm with Adaptive Inertia Weight and Mutation," 2009 World Congress on Computer Science and Information Engineering, 2009, pp. 622-625.

[19] Chiabwoot Ratanavilisagul, Boontee Kruatrachue, "Mutation Period Calculation for Particle Swarm Optimization," 1st International Symposium on Technology for Sustainability, 2011, pp. 213-216.

[20] N. Higashi and H. Iba., "Particle swarm optimization with Gaussian mutation," Proc.of the 2003 IEEE Swarm Intelligence Symphosium, 2003, pp. 72-79.

[21] A. Stacey, M. Jancic, and I. Grundy., "Particle swarm optimization with mutation," Proc. of the 2003 IEEE Congr. On Evol. Comput., 2003, pp. 1425-1430.

[22] D. Dumitrescu, B. Lazzerini, L.C. Jain, A. Dumitrescu, "Evolutionary Computation," CRC Press, 2000.

[23] Hanyong Choi, Shunichi Ohmori, Kazuho Yoshimoto, Hiroaki Ohtake, "Improvement of Particle Swarm Optimization Application of the Mutation Concept for the Escape from Local Minima," 8th International Conference on Supply Chain Management and Information Systems, 2010, pp. 1-5.

[24] KRINK T, VESTERSTROM J S, RIGET J., "Particle swarm optimization with spatial particle extension," Proceedings of IEEE Congress on Evolutionary Computation. Honolulu, Hawaii USA, 2002.

[25] F. van den Bergh., "An Analysis of Particle Swarm Optimizers," PhD thesis, Department of Computer Science, University of Pretoria, South Africa, 2002.