Hybridization of Particle Swarm Optimization with adaptive Genetic Algorithm operators

Suraya Masrom
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
Perak, MALAYSIA
suray078@perak.uitm.edu.my

Irene Moser
Faculty of ICT
Swinburne University
Australia
imoser@swin.edu.au

James Montgomery
Research School of Computer Science
Australian National University
Australia
james.montgomery@anu.edu.au

Siti Zaleha Zainal Abidin
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
Shah Alam, MALAYSIA
sitizaleha533@salam.uitm.edu.my

Nasiroh Omar
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
Shah Alam, MALAYSIA
nasiroh@tmsk.uitm.edu.my

***Abstract*—Particle Swarm Optimization (PSO) is a popular algorithm used extensively in continuous optimization. One of its well-known drawbacks is its propensity for premature convergence. Many techniques have been proposed for alleviating this problem. One of the alternative approaches is hybridization. Genetic Algorithms (GA) are one of the possible techniques used for hybridization. Most often, a mutation scheme is added to the PSO, but some applications of crossover have been added more recently. Some of these schemes use adaptive parameterization when applying the GA operators. In this work, adaptively parameterized mutation and crossover operators are combined with a PSO implementation individually and in combination to test the effectiveness of these additions. The results indicate that an adaptive approach with position factor is more effective for the proposed PSO hybrids. Compared to single PSO with adaptive inertia weight, all the PSO hybrids with adaptive probability have shown satisfactory performance in generating near-optimal solutions for all tested functions.**

***Keywords*-Hybridization; Particle Swarm Optimization; Genetic Algorithm;Adaptive;Crossover;Mutation;**

I. Introduction

Hybridizing Particle Swarm Optimization (PSO) [1] with Genetic Algorithms (GA) [2] has been shown to be an effective method for solving many kinds of optimization problems [3]. The motivation of the hybridization is to alleviate the limitations of PSO with the diversification aspects of GA. Although PSO is very effective in providing results quickly, its ability to find optimal solutions especially for real life problems is still insufficient [4]. Most practical problems are multi-modal and, due to its fast convergence to a single point, PSO tends to converge to a local optimum. This premature convergence problem has been acknowledged many times, and the earliest approach to counteract it is the introduction of the inertia weight factor that influences particle velocity [5]. Even after this measure, many researchers agree that PSO does not have a sufficient exploration ability [6], [7]. Many more methods were introduced to remove or alleviate swarm stagnation [8].

GAs, while still susceptible to premature convergence, are generally found to have better exploration properties than PSO [9], [4]. GAs have several operators that can control the exploration and exploitation search projection: mutation, crossover and selection. Mutation is generally thought to enable exploration, whereas both exploratory and exploitative aspects are ascribed to crossover.

The selection operator enhances exploitation through a reduction of the population. A good balance between exploration and exploitation is considered critical for the success of the search in a multi-modal environment [10]. Explorative aspects introduce diversity into the search direction such that vast areas of the search space can be covered, while exploitation provides the necessary search intensity to optimise the discovered solutions locally.

Apart from the hybridization of algorithms, adaptive parameter control has shown promising results with many algorithms, PSO and GA among them. Whereas on the PSO side, the inertia weight parameter has been widely controlled adaptively [11], on the GA side, adaptive crossover and mutation rates have been demonstrated to produce improved results for many kinds of optimization problem [12]. In this paper, our intention is to explore the strengths of both hybridization and adaptive strategies in improving PSO performance. The research focuses on combining inertia weight PSO with adaptive crossover and mutation rates in an attempt to identify the most effective adaptive approach for PSO hybrid.

The remaining content of this paper is organized as follows. In Section II, brief literature on adaptive PSO and PSO-GA hybridization are given. The proposed PSO hybrid algorithms are described in Section III. Section IV discusses the results before the concluding remarks in Section V.

II. Background

A. Adaptive PSO

Considerable research efforts have been invested in improving the original PSO algorithm. Among other enhancements, adaptive parameterizations of PSO have received much attention from many researchers [13], [7]. Adaptive parameterization periodically adjusts PSO parameters such as inertia weight[13], [7], [14], [15], [11], acceleration coefficients [14], [16] and population size [17], [18] accordingly to the performance of the search. The vast majority of these approaches adjusts the inertia weight parameter. We only report the most relevant papers with 30-200 citations published 2006 and later to capture the most relevant progress on the topic made to date. The most promising of these approaches are applied in the experiments reported later in the paper.

Qin et al. [15] proposed the Individual Search Ability (ISA) scheme, which is a ratio of the distance between the current position and the personal best and the distance between the personal and global bests as shown in Eq. 1.

$$ISA_{i(d)} = \frac{\left|x_{i(d)} - pbest_{i(t)}\right|}{\left|pbest_{i(t)} - gbest_t\right| + \varepsilon} \tag{1}$$

where $x_{i(d)}$ is the position of the ith particle in the dth dimension. $pbest_{i(t)}$ is the personal best position of the ith particle in the tth iteration while $gbest_t$ is the current global best position of the whole swarm. ε is a positive constant close to zero.

A small ISA value indicates that the particle has a weak exploration ability and the inertia weight value ought to be increased, while a larger ISA suggests the particle is searching widely and the weight should be decreased. Eq. 2 implements these principles.

$$w_{i(d)} = 1 - \alpha \left(\frac{1}{1 + e^{-ISA_{i(d)}}}\right) \tag{2}$$

Yang et al. [19] introduced the speed and aggregation measures to monitor the performance of the swarm.The speed measure is calculated according to the personal best fitnesses of the ith particle for the over the last two iterations t and $t-1$ as defined in Eq. 3.

$$speed_{i(t)} = \left|\frac{min(pbestfit_{i(t-1)}, pbestfit_{i(t)})}{max(pbestfit_{i(t-1)}, pbestfit_{i(t)})}\right| \tag{3}$$

The aggregation speed is measured according to the personal and best fitnesses as defined in Eq. 4, where $bestfitness_t$ is the current best fitness found by the particles in the tth iteration and $\overrightarrow{Avg(pbestfit)}_t$ is the average personal fitness of all particles at the tth iteration.

$$aggregation_{i(t)} = \left|\frac{min(bestfitness_t, \overrightarrow{Avg(pbestfit)}_t)}{max(bestfitness_t, \overrightarrow{Avg(pbestfit)}_t)}\right| \tag{4}$$

Both the speed and aggregation measures are used to determine the inertia weight as expressed in Eq. (9):

$$w_{i(t)} = w_{initial} - \alpha \times \left(1 - speed_{i(t)}\right) \times \beta \times aggregation_{i(t)} \tag{5}$$

where α and β are numbers in the range of [0,1].

Arumugam, Senthil and Rao [9] proposed to adjust the inertia weight according to the ratio of global best fitness and mean of personal best fitness to the tth iteration (Eq. 6).

$$w_{i(t)} = 1 - \frac{gbestfit_t}{\overrightarrow{Avg(pbestfit)}_t} \tag{6}$$

Panigrahi, Ravikumar and Das [16] used the best fitness rank of the particle to adapt its inertia weight. The fitnesses are ranked by dimension for the entire population and the resulting rank determines the inertia weight $w_{i(t)}$ of particle i at time t.

$$w_{i(t)} = w_{min} + (w_{max} - w_{min}) \times \frac{Rank\left(pbestfit_{i(t)}\right)}{n} \tag{7}$$

where n is the number of particles and $Rank_{i(t)}$ is the sequence position of particle i based on its personal fitness at the tth iteration.

B. Hybridization of PSO and GA

Acknowledging the advantages of GAs in terms of diversity maintenance, many hybridization methods in the literature have combined GA operators with PSO. A comprehensive survey was provided by Masrom et al. [20]. The majority of these approaches incorporate a version of the mutation operator into a PSO algorithm. The ensuing increase in diversity has been found beneficial by many authors, see for example Achtnig [21] or Esmin et al. [22]. The latter added a uniform random number in the range of $\frac{1}{10}$ of the valid range to the dimensions of a randomly chosen particle. Higashi and Iba [23] incorporated Gaussian mutation into the 'canonical' PSO by Kennedy and Eberhart [1]. Stacey, Jancic and Grundy [24] added Cauchy mutation to the 'canonical' PSO as well as a constriction PSO by Clerc [25]. A comprehensive study by Andrews [26] compared the performance when using Cauchy and Gaussian mutations in PSO implementations with the results when using a special mutation conceived by Michalewicz [27]. The mutation operator chooses a vector randomly, then decides uniformly randomly whether to increase or decrease a dimension's value. If the value is to be increased, a number chosen from the range between the current value and an upper bound is added. The upper bound depends on the limit of the feasible range but is moderated by the current iteration such that the range decreases with increasing iterations and the changes to the variables approach zero over time. Analogously, if the variable is to be decreased, the number to be added is chosen in the range between the current value and the lower bound. The mutation operator introduced by Gao and Xu [4] applies the Henon map distribution. Ting el al. [28] experimented with eight mutation functions, based on differences between numbers, exponents, logarithms and means. All mutations were applied to the $pbest$ individuals only. The results appear to be mixed, with different formulations providing best results for different functions. In general, all authors observe that using mutation, regardless of implementation, improves the performance of PSO.

While Michalewicz's approach [27] decreases the impact of mutation over time, the method does not adjust the mutation effect based on algorithm performance. Adaptive mutation uses particle fitness or diversity as a basis for adjusting the effect of mutation. Alireza [29] used Gaussian mutation with a variable step size that depends on the current best fitness. Zhou and Tan [30] extended the Standard PSO (SPSO) by Bratton and Kennedy [31] where each particle has only two neighbours, by introducing a mutation scheme where uniform mutation is triggered when a swarm becomes 'unhealthy', which is defined as less than $P\%$ of the particles updating their personal best in an iteration. Similarly, Andrews [26] applied mutation only when the swarm was observed to be stagnant, with the $gbest$ of the population remaining unchanged for a number of iterations. All experiments in these studies showed that the adaptive mutation greatly strengthened the global exploration in PSO. Andrews [26] also found that the mutation rate and its dynamic value have a relatively large influence on the performance of the PSO hybrid.

PSO hybrids with a crossover operator have also been shown to outperform 'pure' PSO implementations. Pant and Thangaraj [32] as well as Wang et al. [33] used quadratic crossover but proposed different techniques for selecting and replacing particles. Pant and Thangaraj only chose a particle with the worst position for crossover while Wang et al. performed a conditional replacement of the current particle. In Hao, Wang and Huang's work [34], a simple random crossover with the globally best particle was carried out on each new particle after an update. Only one of the dimensions was swapped between the current particle and the global best. Chen [35], on the other hand, updates all dimensions with aspects of $pbest$ and $lbest$ after a predefined number of iterations in order to include a form of elitism in the algorithm. Park et al. [36] applied a uniformly random probability of swapping values of the particle vector with the $pbest$ at predefined iterations.

Compared to the number of hybrid PSO implementations with adaptive mutation operator, relatively few approaches have incorporated adaptive crossover. Yang et al. [37] introduced several formulations for adaptive crossover which depend on the iteration number. Pant and Thangaraj [32] used particle position and swarm size for measuring the swarm diversity before applying the crossover and PSO update operations. None of the approaches we are currently aware of have used an adaptive probability for crossover. Some studies have included both crossover and mutation in PSO [38], [39], [40], [41], [42], but none of them used adaptive parameterization.

III. The proposed PSO Hybrids

In this study, the PSO hybrids are divided into three algorithms. The first algorithm combines both adaptive parameterizations to the crossover and mutation as illustrated in Fig. 1. The second and third algorithms use only one of the operation.

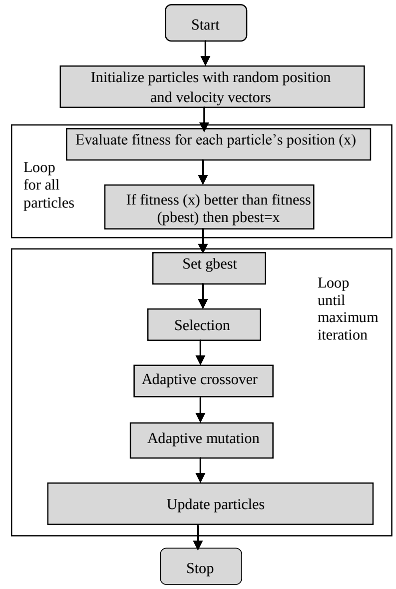

Fig. 1: PSO hybrid with adaptive parameterizations

The algorithm performs crossover between two particles which are chosen probabilistically in proportion to their fitnesses. Each position between the first and the maximum are then swapped according to the adaptive crossover probability C_p. This crossover operator was originally inspired by the $pbest$ crossover introduced by Chen [35] which attempts to increase explorative search in PSO. When the distance between particles and their $pbest$ approaches zero, the particles stop moving and convergence sets in. Therefore, the $pbest$ crossover tends to reduce PSO convergence by moving the position of a particle away from its $pbest$ [35].

Unlike the original $pbest$ crossover by Chen [35], which uses periodic crossover, the current implementation applies crossover according to an adaptive crossover probability.

The threshold value r is set to a random number in the interval $[0, 1]$. It is compared to each particle's probability C_p of crossover to decide whether this particle's randomly chosen position d should be altered using $pbest$ crossover. As proposed in Chen [35], the adjustment to dimension d is made using an average of two particles' relevant $pbest$ values.

In the algorithmic listing 1, the crossover probability C_p of all particles is adapted in line 2 using one of the approaches Qin et al. [15], Yang et al. [19], Arumugam, Senthil and Rao [9] and Panigrahi, Ravikumar and Das [16] as described in Section II-A.

The mutation operation uses the adaptive mutation probability M_p similar to the crossover probability C_p. Eq. 8 defines the mutation operation.

$$X_{i(d)} = X_{i(d)} + Gaussian(\alpha) \quad (8)$$

where the $Gaussian(\alpha)$ is a Gaussian function that returns a random value from the range of the particle dimension. The α

Algorithm 1: Adaptive Crossover

foreach *particle* $x_i \in population$ **do**
  calculate adaptive crossover probability C_p
set r to a uniform random number $[0, 1]$;
set d to a uniform random integer $[0, dim]$;
choose $pbest1, pbest2$ uniformly randomly among all n particles;
foreach *particle* $x_i \in population$ **do**
  if $r < C_p$ **then**
    $x_{id} = x_{id} + \frac{pbest1_d + pbest2_d}{2}$;

value is bounded within 0.1 times of the particle dimension.

IV. Experiments

The objective of the experiments serves two purposes. The first is to identify the most effective of four state-of-the-art adaptive approaches for PSO hybrids. The second is to compare the performance of adaptive PSO hybrids and adaptive inertia weight PSOs. The experiments are divided into four sets which are adaptive crossover rate (ACR), adaptive mutation rate (AMR), adaptive crossover and mutation rate (ACMR) and adaptive inertia weight without hybridization (AIW). Initially, each set is combined with each of the four adaptive interia weights. The most successful combination is subsequently used for a comparison between ACR, AMR and ACMR.

Each set of experiments was repeated 30 times with 2000 iterations. Therefore, regardless of algorithm, each of the 30 trials was allowed an equal number of 80000 function evaluations. The general experiment setting that relates to all algorithms is given in Table I.

TABLE I: General experiment setting

Attribute	Value
Number of particles, *n*	40
Particle dimension, *dim*	30
Personal learning rate, c_1	0.9
Social learning rate, c_2	0.9

A. Adaptive Approach

In order to identify the most suitable adaptive approach for the different PSO hybrids, we have tested the adaptive approaches for inertia weight described in Section II-A. The approaches have several adaptive factors relating to current PSO performance as given in Table II.

In terms of the adaptive approaches, the best and global fitnesses are two measures of achievement in PSO. The best fitness is the current best fitness found by the particles in a particular iteration while the global fitness presents the optimal value that the swarm has found up to a certain number of iterations. Based on preliminary experiments, the α and β parameters were set to 0.4 and 0.8 respectively for the first approach while in the fourth approach, the ε and α are configured as 0.9 and 0.3 respectively. Besides, the third approach has minimum and maximum attributes specifically for each adaptive parameter. The minimum value is used as initial configuration for each adaptive parameter of the second approach. The configuration of these attributes are listed in Table III.

TABLE II: Adaptive approaches

Approach	Adaptive factor	Researcher.
Adaptive 1	Personal fitness Best fitness	Yang et al.[19]
Adaptive 2	Global fitness Personal fitness	Arumugam and Rao[9]
Adaptive 3	Personal fitness	Panigrahi, Ravikumar and Das [16]
Adaptive 4	Particle position	Qin et al.[15]

TABLE III: Parameter configuration for third approach

Adaptive parameter	Attribute	Value
Crossover rate	Minimum, c_{min} Maximum,c_{max}	0.1 1.0
Mutation rate	Minimum,m_{min} Maximum,m_{max}	0.4 0.9
Inertia weight	Minimum,w_{min} Maximum,w_{max}	0.4 0.9

B. Benchmark Functions

To illustrate the effectiveness of the PSO hybrids, a set of six well-known benchmark functions are employed, with the Sphere function denoted as f_1, the Rosenbrock function as f_2, the Rastrigin function as f_3, the Levy function as f_4, Griewank as f_5 and Ackley's functions as f_6. These benchmark functions have optimal values of zero.

V. Results and Discussion

The results are divided into two parts. Firstly, the most suitable adaptive approach for PSO hybrids (ACR, AMR and ACMR) is determined. Then, the performance of PSO with the selected adaptive approach is compared among different PSO algorithms (ACR, AMR, ACMR and AIW).

A. Comparison of Adaptive Approaches

Fig. 2 shows the comparative performances of ACR, AMR and ACMR using each of the four adaptive approaches on the six test functions. Across all adaptive algorithms, the adaptive approach by Qin et al. [15] is the most effective. The approach uses the ratio between the particle's distance to its $pbest$ and the distance of its $pbest$ to $gbest$ instead of fitness-based factors. Especially in the case of adapting crossover (encompassing both the ACR and ACMR settings), adaptive approach 4 outperforms the others on all objective functions. The results when adapting the mutation rate in the AMR setting are not as clearly in favour of method 4, but it outperforms the others on three in six functions. Based on these results, we conclude that a particle's position in

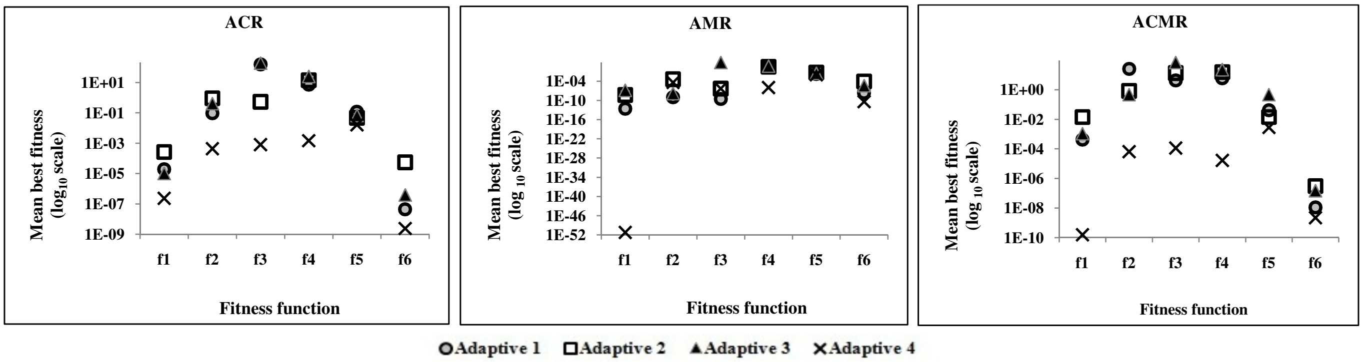

Fig. 2: PSO hybrids with four different adaptive approaches tested on all functions (f1-f6)

relation to its personal best position is highly relevant when determining the stage of the search, at least in the search space of the six functions optimised here.

The results regarding the other three adaptation methods are mixed. Method 1 performs better than the other two in 10 cases of a total of 16. While methods 2 and 3 are based on the current best fitness values or their ranks, method 1 includes an aspect of fitness development over the last iteration which seems to be beneficial for the performance of the algorithm. However, it cannot compare with method 4 on the functions used for these experiments.

B. Comparison of PSO Hybrids

The performance of PSO with the fourth adaptive approach is compared among the PSO hybrids (AMR, ACR, ACMR) and single PSO with inertia weight (AIW). Table IV shows the mean best fitness, standard deviation and mean number of iterations until the best solution was found of all experiments. Mean best fitness indicates the accuracy of results while number of iterations is used to measure the efficiency of each algorithm to converge and achieve near-optimal solutions.

In general, all PSO algorithms with adaptive approach have achieved good results for all tested functions with very low values for the mean best fitness. The results in Table IV show that all adaptively parameterized hybrids mostly outperform inertia weight PSO. However, hybridizing PSO with adaptive crossover does not appear to have the expected benefit. Although ACR mostly outperforms inertia weight PSO, except in the case of f_2, it never performs as well as the mutation-based hybrid AMR. Including both mutation and crossover (ACMR) improves on the performance of the hybrid with crossover (ACR) in four of the six experiments. The number of iterations used indicates when the algorithm has stagnated. Usually producing worse results takes fewer iterations, but in some cases AMR produces better results faster than the other variations take to produce lower quality results.

VI. Conclusions

According to our knowledge on the state of the art of PSO, adaptive parameterization and hybridization can make significant improvements regarding the algorithm's performance.

TABLE IV: The mean best fitness (standard deviation) and mean number of iterations to achieve the best solutions over 30 runs for each setting.

f	ACR	AMR	ACMR	AIW
f_1	2.34E-07 (1.65E-05) 1702	**6.06E-52** (4.76E-48) 1997	1.54E-10 (2.63E-05) 1865	6.88E-08 (7.66E-06) 956
f_2	4.35E-04 (3.20E-02) 833	**3.68E-05** (8.40E-04) 1903	6.47E-05 (1.93E-05) 1074	2.58E-04 (5.94E-01) 1630
f_3	8.08E-04 (2.15E-05) 1502	**4.93E-07** (1.15E-09) 1397	1.14E-02 (1.40E-02) 1660	1.07E-02 (2.56E-01) 1020
f_4	1.46E-03 (5.53E+00) 1644	**1.05E-06** (1.44E+01) 1388	1.66E-05 (2.85E+01) 1396	3.88E-05 (7.83E+00) 794
f_5	1.64E-02 (3.15E-01) 1862	**1.02E-03** (2.21E-01) 1918	2.80E-03 (2.73E-04) 1825	2.96E-02 (1.88E-01) 812
f_6	2.40E-09 (7.66E-11) 1517	**3.91E-11** (5.74E-17) 1528	2.20E-09 (2.88E-07) 1533	6.99E-09 (5.87E-05) 947

Most approaches, however, separate the two approaches. In this study we have introduced a number of PSO hybrids combined with adaptive parameterization.

In comparison to PSO with adaptive inertia weight, most variations of PSO hybrids with adaptive parameterization have produced better results. However, among the PSO hybrids, the best optimal results have been achieved by the inclusion of adaptive mutation. A combination of both crossover and mutation in PSO has consistently led to better results than the inclusion of crossover on its own, whereas it performs consistently worse than including mutation on its own. The mutation aspect therefore appears to be the driving force behind the improvement.

Regardless of the type of hybridization, the most effective adaptive parameterization method is based on differences in positions between the current, $pbest$ and $gbest$ solutions rather than their fitnesses. Therefore, further investigations into the exploitation of position information appear to be a promising research area in adaptive parameterization.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[2] M. Affenzeller, S. Wagner, S. Winkler, and A. Beham, *Genetic algorithms and genetic programming: modern concepts and practical applications*. CRC Press, 2009.

[3] X. Shi, Y. Lu, C. Zhou, H. Lee, W. Lin, and Y. Liang, "Hybrid evolutionary algorithms based on PSO and GA," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03*, vol. 4. IEEE, 2003, pp. 2393–2399.

[4] H. Gao and W. Xu, "Particle swarm algorithm with hybrid mutation strategy," *Applied Soft Computing*, vol. 11, no. 8, pp. 5129–5142, Dec. 2011.

[5] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence*. Wiley, 2005.

[6] C.-Y. Chen, K.-C. Chang, and S.-H. Ho, "Improved framework for particle swarm optimization : Swarm intelligence with diversity-guided random walking," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12 214–12 220, Sep. 2011.

[7] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft computing*, 2011.

[8] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. part i: background and development," *Natural Computing*, vol. 6, pp. 467–484, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11047-007-9049-5

[9] M. S. Arumugam and M. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing*, vol. 8, no. 1, pp. 324–336, Jan. 2008.

[10] K. C. Tan, S. C. Chiam, A. A. Mamun, and C. K. Goh, "Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization," *European Journal of Operational Research*, vol. 197, no. 2, pp. 701–713, Sep. 2009.

[11] K. Suresh, S. Ghosh, D. Kundu, A. Sen, S. Das, and A. Abraham, "Inertia-Adaptive particle swarm optimizer for improved global search," in *Eight International Conference on Intelligent System Design and Applications-ISDA2008*, 2008.

[12] W.-Y. Lin, W.-Y. Lee, and T.-P. Hong, "Adapting crossover and mutation rates in genetic algorithm," *Journal of Information Science and Engineering*, vol. 19, pp. 889–903, 2003.

[13] A. B. Hashemi and M. R. Meybodi, "A note on the learning automata based algorithms for adaptive parameter selection in pso," *Applied Soft Computing*, vol. 11, no. 1, pp. 689 – 705, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494609002877

[14] T. Niknam, "A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem," *Applied Energy*, vol. 87, no. 1, pp. 327 – 339, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030626190900213X

[15] Z. Qin, F. Yu, Z. Shi, and Y. Wang, "Adaptive inertia weight particle swarm optimization," in *Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg, 2006, pp. 450–459.

[16] B. K. Panigrahi, V. Ravikumar Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy Conversion and Management*, vol. 49, no. 6, pp. 1407–1415, Jun. 2008.

[17] D. Chen and C. Zhao, "Particle swarm optimization with adaptive population size and its application," *Applied Soft Computing*, vol. 9, no. 1, pp. 39–48, Jan. 2009.

[18] L. W-F and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 5, pp. 1270 –1293, Oct. 2008.

[19] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1205–1213, 2007.

[20] S. Masrom, S. Z. Zainal Abidin, P. N. Hashimah, and A. S. Abd Rahman, "Low-level teamwork hybridization for p-metaheuristics: A review and comparison," in *2011 3rd Conference on Data Mining and Optimization (DMO)*, June 2011, pp. 128 –133.

[21] J. Achtnig, "Particle swarm optimization with mutation for high dimensional problems," *Studies in Computation Intelligence (SCI)*, vol. 82, pp. 423–439, 2008.

[22] A. A. A. Esmin, G. Lambert-Torrs, and G. B. Alvarenga, "Hyrid evolutionary algorithm based on PSO and GA mutation," in *IEEE Sixth International Conference on Hybrid Intelligent Systems*, 2006, pp. 57–63.

[23] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation," in *IEEE Swarm Intelligence Symposium- SIS '03.*, 2003, pp. 72–79.

[24] A. Stacey, M. Jancic, and I. Grundy, "Particle swarm optimization with mutation," in *The 2003 Congress on Evolutionary Computation, 2003 -CEC '03.*, vol. 2, 2003, pp. 1425–1430.

[25] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58 –73, Feb 2002.

[26] P. S. Andrews, "An investigation into mutation operators for particle swarm optimization," in *Evolutionary Computation, 2006*, 2006, pp. 1044–1051.

[27] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, 1996.

[28] T. O. Ting, M. V. C. Rao, C. K. Loo, and S. S. Ngu, "A new class of operators to accelerate particle swarm optimization," in *IEEE Congress on Evolutionary Computation*, vol. 4, 2003, pp. 2406–2410.

[29] A. Alireza, "PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems," *Acta Automatica Sinica*, vol. 37, no. 5, pp. 541–549, 2011.

[30] Y. Zhou and Y. Tan, "Particle swarm optimization with triggered mutation and its implementation based on GPU," in *12th annual conference on Genetic and evolutionary computation - GECCO '10*, 2010, pp. 1–8.

[31] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, April 2007, pp. 120 –127.

[32] M. Pant and R. Thangaraj, "A new particle swarm optimization with quadatic crossover," in *15th International Conference on Advanced Computing and Communications*, 2007, pp. 81–86.

[33] L. Y. Wang H, Wu Z and Z. S, "Particle swarm optimization with a novel multi-parent crossover operator," in *Fourth International Conference on Natural Computation*, 2008, pp. 664–668.

[34] Z.-F. Hao, Z.-G. Wang, and H. Huang, "A particle swarm optimization algorithm with crossover operator ," in *The 6th International Conference on Machine Learning and Cyernetics*. Hong Kong: IEEE, 2007, pp. 1036–1040.

[35] S. Chen, "Particle swarm optimization with pbest crossover," in *The IEEE Congress on Evolutionary Computation-CEC2012*, Brisbane, Australia, 2012, pp. 57–63.

[36] J.-B. Park, Y.-W. Jeong, J.-R. Shin, K. Lee, and J.-H. Kim, "A hybrid particle swarm optimization employing crossover operation for economic dispatch problems with valve-point effects," in *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, Nov. 2007, pp. 1 –6.

[37] D. Yang, J. Chen, and M. Naofumi, "Time-varying mutation in particle swarm optimization," in *IEEE Third International Conference on Natural Computation*, vol. 4, 2012, pp. 160–164.

[38] Y. Kao and E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions," *Applied Soft Computing*, vol. 8, no. 2, pp. 849 – 857, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494607000683

[39] Y. Marinakis and M. Marinaki, "A hybrid genetic − particle swarm optimization algorithm for the vehicle routing problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1446 – 1455, 2010.

[40] Y. Shao, Q. Chen, and C. Li, "Research on hybrid improved PSO algorithm," in *Computational Intelligence and Intelligent Systems*, ser. Communications in Computer and Information Science, Z. Cai, H. Tong, Z. Kang, and Y. Liu, Eds. Springer Berlin Heidelberg, 2010, vol. 107, pp. 234–242. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16388-3_26

[41] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Information Processing Letters*, vol. 93, no. 5, pp. 255–261, 2005.

[42] W. Zhong, J. Xing, and F. Qian, "An improved theta-pso algorithm with crossover and mutation," in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, June 2008, pp. 5308 –5312.