Optimal Capacitor Placement Using Hybrid PSO and HBMO Algorithm

Mousa Yousefzadeh *, Hamed Ahmadi, Ahmad Rostamian , Ayoob Asadi

Department of Electrical Engineering, Damghan Branch, Islamic Azad University, Damghan, Iran
m.zaher.ghorbani@gmail.com

Abstract: This paper presents a new and efficient approach for capacitor placement in transmission systems that determine the optimal locations and number of capacitor with an objective of improving the voltage profile and reduction in power loss. The solution methodology has two parts: in part one determine the number and size of capacitor and in part two a new hybrid algorithm is used to estimate the optimal bus of capacitors at the optimal size and number at part one. This algorithm is PSO and HBMO hybrid. The main advantage of the proposed method is that it faster than other methods. The proposed method is applied to 14-bus IEEE Transmission system. The solutions obtained by the proposed method are compared with PSO method. The proposed method has outperformed the other methods in terms of the quality of solution (Convergence speed and size of the objective function).

[Yousefzadeh M, Ahmadi H, Rostamian A, Asadi A. **Optimal Capacitor Placement Using Hybrid PSO and HBMO Algorithm.** *J Am Sci* 2013;9(7):167-174]. (ISSN: 1545-1003). http://www.jofamericanscience.org. 21

Keywords: loss reduction, transmission systems, HBMO, PSO

INTRODUCTION

For any research on electrical energy system required We calculate parameters such as voltage buses, producing power generators, and system losses. Network of 14 bus IEEE system is a system for transmission network and method used for load flow in the system is newton raphson[1]. A method for optimizing a criterion of superiority over other methods used are the better objective function value that is less and the other is the convergence speed. In recent years, various methods for optimizing Placement capacitors is presented. References [10] and [11] in a good way to find the optimal size and location suitable capacitor is presented. More recently, researchers thought for a coalition of innovative and intelligent techniques with conventional methods have failed to resolve some issues. This act to strengthen and improve the highly mathematical models and numerical methods are. Plus the efficiency calculations are also done to maintain this action. Good framework for the evolution of birds, so that theory is considered. In this paper we consider the improvement in voltage profiles and reduce system losses, as the problem of mathematical modeling and algorithm is solved using a hybrid pso and hbmo.

PROBLEM FORMULATION

The objective of capacitor placement in the transmission system is to minimize the loss and improve voltage profile. For simplicity, the operation and Calculation of the capacitor placed in the transmission system assume that three-phase system is considered as balanced and loads are assumed as time invariant. Mathematically, the objective function of the problem is minimizing the loss and voltage deviation. This function is as 1:

$$F = W_1 \times p_{loss} + W_2 \times \sum_{i=1}^{14}(1 - v_i)^2 \qquad (1)$$

Where w_1 and w_2 are the objective function coefficient of weight loss and voltage deviation. P_{loss} and v_i are the total loss in transmission system and the voltage magnitude of bus i.

The voltage magnitude at each bus must be maintained within its limits and is expressed as:

$$v_{min} < |v_i| < v_{max} \qquad (2)$$

Where $|v_i|$ is the voltage magnitude of bus i, v_{min} and v_{max} are bus minimum and maximum voltage limits, respectively.

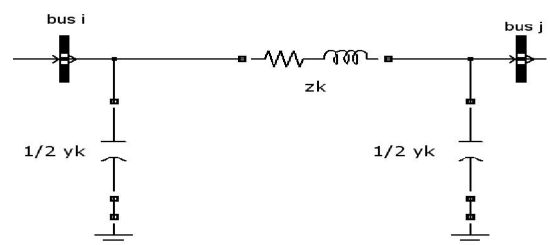

Figure1: pi model of transmission line between buses

The power flow is computed by following equation:

$$p_i = |v_i| \sum_{j=1}^{n} |y_{ij}| |v_j| \cos(\delta_i - \delta_j - \emptyset_{ij}) \quad (3)$$

$$Q_i=|v_i|\sum_{i=1}^{n}|y_{ij}||v_i|\sin(\delta_i\ \delta_j\ \emptyset_{ij}) \quad (4)$$

Where p_i and Q_i are the real and reactive power flowing out of bus i. $|y_{ij}|$ and $\emptyset_{ij}$ are the Size and angle of impedance. δ_i And δ_j are the angle of voltage at bus i and j.

$$I_{ij}=\frac{v_i-v_j}{z_k}-\frac{1}{2}y_k v_i \quad (5)$$

Where z_k and y_k are the admittance and impedance line k. $I_{ij}$are the current from bus i to bus j.

$$s_{ij}=v_i I_{ij}^*$$

$$s_{ji}=v_j I_{ji}^*$$

$$s_{lossk}=s_{ij}-s_{ji}$$

Then

$$p_{lossk}=real(s_{lossk})$$

$$p_{losst}=\sum_{k=1}^{n}p_{lossk}$$

PSO and HBMO algorithm

First describe pso and hbmo algorithm then describe hybrid pso and hbmo algorithm.

PSO algorithm

Kennedy and Eberhart [2], considering the behavior of swarms in the nature, such as birds, fish, etc. developed the PSO algorithm. The PSO has particles driven from natural swarms with communications based on evolutionary computations. PSO combines self-experiences with social experiences. In this Algorithm, a candidate solution is presented as a particle. It uses a collection of flying particles (Changing solutions) in a search area (current and possible solutions) as well as the movement towards a promising area in order to get to a global optimum.

$$prtivel_i^j = w\times prtivel_i^{j-1} + c_1\times rand\times(pbest_j^{i-1}-prtpos_j^{i-1}) + c_2\times rand \times(gbest_j^{i-1}-prtpos_j^{i-1})$$

When:

c_1, c_2 : The balance factors between the effect of self-knowledge and social knowledge in moving the particle towards the target. (Usually the value 2 is suggested for both factors in the literature)

$Rand$: A random number between 0 and 1, and different at each iteration

W : Inertia weight

$prtivel_i^j$: The velocity of jth particle in ith iteration

$pbest_j^i$: The best position of a particle

$gbest_j^{i-1}$: The best position within the swarm

$prtivel_i^j$: The velocity of jth particle in ith iteration

$prtpos_j^i$: The position of jth particle in ith iteration

Since, in the above algorithm, there is the possibility of particles movement to out of the problem space [3], an upper velocity bound for particle movement is specified. One of the PSO problems is its tendency to a fast and premature convergence in mid optimum points [9]. A lot of effort has been made so far to solve this problem. For instance, in [4] the best value for w in (1) is set to 0.9, which linearly decreases to 0.4.

MAJOR PSO-BASED ALGORITHMS

2-D Otsu PSO (TOPSO) – This algorithm is a combination of the PSO and the optimal threshold selecting search in order to improve the PSO performance [12].

Active Target PSO (APSO) –

In this algorithm, in addition to two existing terms; namely the best position and the best previous position for particle velocity updating, a third term called ‘Active target is also utilized. Calculating the third term is complicated and it does not belong to the existing positions. This method maintains the diversity of the PSO as well as not trapping in the local optimum [13].

Adaptive PSO (APSO) –

During the running process of the PSO, sometimes a number of particles are inactive, that is, they do not have the ability of local and global searching and do not change their positions a lot, so their velocity is nearly reached to zero. One solution is to adaptively replace the current inactive particles with fresh particles in a way that the existing PSO-based relationships among the particles are kept. This is done by APSO method [14].

Adaptive Mutation PSO (AMPSO) –

This algorithm utilizes the adaptive mutation using Beta distribution in the PSO. It includes two types: AMPSO1 and AMPSO2. The former mutes the best individual position in the swarm and the latter mutes the best global position [15].

Adaptive PSO Guided by Acceleration Information (AGPSO)

This algorithm is for improving the PSO efficiency in finding the global optimum. The acceleration item is also added to position and velocity updating equations and then, convergence analysis is performed [16].

Angle Modulated PSO (AMPSO) –

This algorithm employs a trigonometry function to generate a bit string. Its difference with the Binary PSO (BPSO) algorithm lies in its high computational efficiency. That is, it avoids the generation of high-dimensional binary vector and thus, its discretion process is not complicated. Moreover, it changes all of the high-dimensional problems to a four-dimensional problem. Hence, it saves a large amount of the memory and is easy to run [17].

Attractive Repulsive Particle Swarm Optimization (ARPSO) –

This algorithm has been developed to remove the PSO’s drawback in premature convergence. It includes an attractive phase and a repulsive phase. In the attractive phase, the addition operator is used among the equation terms for velocity updating. In the repulsive phase, the subtraction operator is employed. Indeed, the particles are attracted towards each other in the attractive phase and get away from each other in the repulsive phase [18].

Augmented Lagrangian PSO (ALPSO) –

This algorithm is a combination of Augmented Lagrangian method and the PSO algorithm. It is applied to optimization problems having equal and unequal constraints [19].

Best Rotation PSO (BRPSO) –

This algorithm is used to optimize multimodal functions and in fact, the swarm is divided into several sub-swarms. It is worth mentioning that the swarm separation and its division on several populations do not look reasonable for single modal problems. However in normal PSO in multimodal functions the wide knowledge of the whole population performance make the system converges too fast and also increases the probability of stagnation into local minima but in BRPSO when best rotation is executed, stagnation on local minima is avoided by forcing populations to move from one local minimum to another one, increasing the exploration of the problem space between different local minima. This algorithm is in a way that a periodically rotation is performed among the particles of different sub-swarms [20].

Binary PSO (BPSO) –

The difference between PSO and BPSO lies in their defined searching spaces. In the typical PSO, moving in the space means a change in the value of position coordinates in one or more of existing dimensions. However, in the BPSO moving in the spaces means a change in the probability of the fact that the value of position coordinate is zero or one [21].

Co-evolutionary PSO –

This algorithm was proposed in 2002 in [22].

Combinatorial PSO (CPSO) –

This algorithm is employed to optimize hybrid problems (consisted of continuous and integer variables) [23].

Comprehensive Learning PSO (CLPSO) –

In [24], the new velocity updating function is proposed and employed to construct CLPSO and then the new algorithmis tested using a group of benchmark functions.

Concurrent PSO (CONPSO) –

In 2004, the CONPSO algorithm was developed in [25].

Constrained optimization via PSO (COPSO) –

The COPSO algorithm is applied to constrained single objective problems. In this algorithm, a technique is employed to investigate the constraints and it has an external file, called "Tolerant", to save the particles. Indeed, in this technique some particles are missed through setting constraints. In order to develop the lifetime of these particles, the above-mentioned external file is utilized and a ring topology structure is employed. In fact, the COPSO is a kind of improvement in Lbest version of the PSO. Moreover, the external procedure, which maintains swarm diversity and guidance towards good points keeping the self-setting capacity, are utilized [26].

Cooperative PSO (CPSO_M) –

In 2004, this algorithm was presented in [27] wherein a multi-cooperative algorithm was schemed.

Cooperative PSO (CPSO_S) –

In 2004, this algorithm was schemed in [28] in which a single cooperative algorithm was introduced.

Cooperatively Coevolving Particle Swarms CCPSO –

This algorithm is suitable for large-scale problems. It breaks the problem into some smaller-scaled ones in a way that the internal dependencies of generated particles are in the possible least values. Then, these particles will become cooperated [29].

Cooperative Multiple PSO (CMPSO) –

Since the PSO efficiency when solving multi-dimensional problems is reduced, the CMPSO algorithm is introduced to overcome this problem. This algorithm has all conductivity and control properties of the PSO [30].

Cultural based PSO (CBPSO) –

This algorithm is in fact the use of PSO in cultural algorithm (CA) framework. Because of PSO's drawback in finding the global optimum and on the other hand, the effecting of the CA in finding the global optimum due to having multiple evolutions and multiple progresses, using them simultaneously can enhance the PSO [31].

Dissipative PSO (DPSO) –

Sometimes the evaluation in the PSO becomes static because of swarm's tendency to get the equilibrium status. Thus, the algorithm will be prevented from searching for more areas and it may occasionally be trapped in a local minimum. In order to overcome this problem, a dissipative system is made using the DPSO algorithm introducing the negative entropy and producing craziness among particles. Utilizing of this system will practically prevent the above-mentioned stagnancy [32].

Divided range PSO (DRPSO) –

In this method, wherein there are several objective functions, particles are first divided to sub-swarms based on one of the objective functions value. Next, the discrete PSO algorithm is run in each sub-swarm. If the stop condition is satisfied, the algorithm will finish; otherwise, the particles are gathered again and are ordered based on the next objective function and the categorizing takes place once more. This algorithm is employed for the clustering of hoc and mobile networks [33].

Dual Similar PSO Algorithm (DSPSOA) –

This algorithm is schemed in [34] wherein through the improvement of the option modes of gbest and pbest of the PSO algorithm, an effective dual similar particle swarm optimization algorithm (DSPSOA) is presented.

Dynamic adaptive dissipative PSO (ADPSO) –

In this algorithm, on the one hand, a dissipative is made for the PSO introducing negative entropy and on the other hand, a mutation operator is utilized to increase the variety in the swarm when it reaches an equilibrium condition in last runs. Thus, it generates an adaptive strategy for inertia weight in order to keep the balance between the local and global optimality [35].

Dynamic and Adjustable PSO (DAPSO) –

In order to make a balance between the discovery and extraction in the PSO and also to keep and protect the particles diversity, DAPSO algorithm has been proposed in which the distance of each particle to the best position is calculated to adjust the velocity of particles in each step [36].

Dynamic Double Particle Swarm Optimizer (DDPSO) -

This algorithm, using a convergence analysis, guarantees the convergence to the global optimal solution. Particle position constraints are set dynamically in this method [37].

Dual Layered PSO (DLPSO) –

The DLPSO algorithm is developed to design a neural network. This algorithm optimizes the network in an architectural layer. It is used for neural network joint weights. A classic boost power transformer is employed to test neural network controllers [38].

Dynamic neighborhood PSO (DNPSO) –

The DNPSO method has some modifications to the conventional PSO. In this method, instead of using the current Gbest in the PSO, another parameter, called Nbest, is utilized. This term is the best particle among the current particle's neighbors in a specified neighborhood. This method discusses that the selection of neighbors for the current particle, as an objective, is multi-objective. In addition, the selection of their best is another objective [39].

Estimation of Distribution PSO (EDPSO) –

This algorithm is a hybrid of the PSO and Estimation of Distribution Algorithm (EDA). Indeed, the ED algorithms−using the obtained information from stochastic models upon which good solution areas on distribution are generated during the optimization process−try to find better areas. This feature of such an algorithm is utilized to improve the performance of PSO [40].

Evolutionary Iteration PSO (EIPSO) –

This algorithm is a combination of the PSO and Evolutionary Programming (EP). Thus, it is able to increase the computational efficiency of EP and it can avoid trapping the algorithm in local optimum [41].

Evolutionary Programming and PSO (EPPSO) –

This algorithm is a combination of the PSO and EP. Indeed, the combination of these two algorithms will cause a help for the PSO capability in making a balance between local and global search to the faster convergence of the EP algorithm. On the other hand, the PSO's drawback in lacking diversity among the particles with mutation between elements in the EP is to some extent removed [42].

Extended Particle Swarms (XPSO) –

Using the Genetic Programming, various algorithms driven from the PSO can be obtained in [43].

Extended PSO (EPSO) –

In this algorithm, the contemporary advantages of Gbest and Lbest versions are utilized. In fact, a hybrid of both is employed in velocity updating equation. The difference between these algorithms with the Fully-informed PSO (FIPS) algorithms lies in less computational costs [44].

HBMO algorithm

The honey bee is a social insect that can survive only as a member of a community, or colony. The colony inhabits an enclosed cavity. The honey bee community consists of three structurally different forms-the queen (reproductive female), the drone(male) and the worker (no reproductive female). These castes are associated with different functions in the colony; each caste possesses its own special instincts geared to the needs of the colony. The HBMO Algorithm combines a number of different procedures [5-7]. Each of them corresponds to a different phase of the mating process of the honey bee. A drone mates with a queen probabilistically using an annealing function as follows [5-7]:

$$\mathrm{Prob}(D) = \exp(-D(f)/S(t))$$

where Prob(D) is the probability of adding the sperm of drone D to the spermatheca of the queen (that is, the probability of a successful mating), D(f) is the absolute difference between the fitness of D and the fitness of the queen (for complete description of the calculation of the fitness function see below) and S(t) is the speed of the queen at time t. The probability of mating is high when the queen is with the high speed level, or when the fitness of the drone is as good as the queen's [8]. After each transition in space, the queen's speed decreases according to the following equations:

$$S(t+1) = \propto \times S(t)$$

Where a is a factor Î (0,1) and is the amount of speed and energy reduction after each transition and each step. Initially, the speed of the queen is generated at random. A number of mating flights are realized. At the start of a mating flight drones are generated randomly and the queen selects a drone using the probabilistic rule in Eq. 9. If the mating is successful (i.e., the drone passes the probabilistic decision rule), the drone's sperm is stored in the queen's spermatheca. By using the crossover of the drone's and the queen's genotypes, a new brood (trial solution) is generated, which can be improved later by employing workers to conduct local search. One of the major differences of the HBMO algorithm from the classic evolutionary algorithms is that since the queen stores a number of different drone's sperm in her spermatheca, she can use parts of the genotype of different drones to create a new solution which gives the possibility to have fittest broods more. In real life, the role of the workers is restricted to broodcare and for this reason the workers are not separate members of the population and they are used as local search procedures in order to improve the broods produced by the mating flight of the queen. If the new brood is better than the current queen, it takes the place of the queen. If the brood fails to replace the queen, then in the next mating flight of the queen this brood will be one of the drones.

PROPOSED TECHNIQUE

In this method uses the function prob from hbmo algorithm for Algorithm to speed be faster in fact the function be faster converge.

HBMO method is defined as a function prob:

$$\mathrm{prob}(Q,D) = e^{-\frac{\Delta(f)}{s(t)}}$$

Where s(t) is the particle speed in hbmo method.

$$v_i^{k+1} = wv_i^k + c_1\mathrm{rand}_1 \times (\mathrm{pbest}_i - s_i^k) + c_2\mathrm{rand}_2 \times (\mathrm{gbest} - s_i^k)$$
$$\Delta(f) = \mathrm{gbest} - x_i^k$$

And $\Delta(f)$ Public transportation is the best value. Improved speed and convergence of systems to be faster.

Test result

The proposed method has been programmed using MATLAB and run. The effectivenss of the proposed method for loss reduction and improv voltage profile by capacitor placement test on 14-bus IEEE systems. The results obtained in this method are explained in the following sections.

14-bus IEEE system

In this paper, IEEE 14-bus system is chosen for the case studies. The original system consist of two generators, three synchronous compensators, eleven loads, three transformer, 14 buses and fifth teen lines. The system is shown in figure 2.

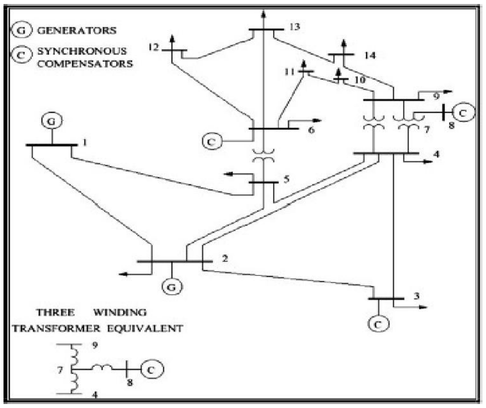

Figure2: IEEE 14-bus system

Ofter system test simulation in MATLAB program can see the voltage profile in the figure 3.

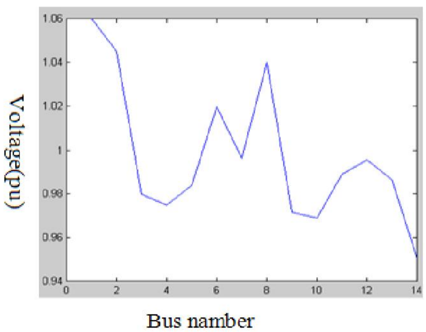

Figure3: voltage profile in 14-bus system

A method for optimizing a criterion of superiority over other methods are used Include an amount is less than that is the objective function and the convergence rate, which is an expression of what is fast achieving further integration will be less than this amount. These two parameters determine which of the five-fold the number of capacitors in the system, we compare the results. In the first part of the objective function value and the second section we discuss the convergence speed.

Value of the objective function

For this purpose, the following table that corresponds to the number of capacitors, the size and type of objective function is to form.

Table1: objective function value for two method optimization

	one capacitor	two capacitor	Three capacitor	Four capacitor	Five capacitor
PSO	9.3326	9.2995	9.2932	9.3039	9.3146
PSO& HBMO	9.3326	9.2995	9.2932	9.3044	9.3278

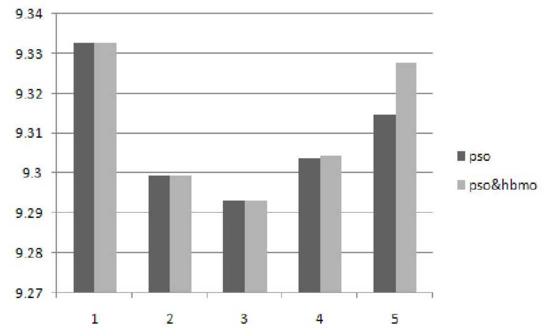

Figure4: Diagram for value of two methods

As is clear from the above chart table and increasing the number of capacitors using the objective function is pso optimum value more than pso & hbmo method is superior.

Convergence speed

Table2: The convergence speed of PSO and PSO & HBMO methods

	one capacitor	two capacitor	Three capacitor	Four capacitor	Five capacitor
PSO	2	3	11	14	20
PSO& HBMO	1	2	7	8	9

As is clear from the above table and chart combination method is faster than the PSO method. The reason is that the combined method of Prob pan HBMO method is used to improve speed.

Comparison of PSO and PSO & HBMO

Comparison of two parts before it can be concluded that the convergence speed is important used combined mathode, and if the value of the objective function is important pso method is used.

*** Corresponding Author:**
Mousa Yousefzadeh
Department of Electrical Engineering
Damghan Branch, Islamic Azad University
Damghan, Iran
m.zaher.ghorbani@gmail.com

REFERENCES

[1] Sadat, H., Power system Introduction, [2] Kennedy, J and Eberhart, R.C., 1995. Particle Swarm Optimization. IEEE Int. Conf. on N.N, Proc: 1942-1948.
[3] Eberhart, R.C., Simpson, P.K and Dobbins, R.W., 1996. Computational Intelligence PC tools. 1st ed. Boston, MA: Academic press professional.
[4] Eberhart, R.C and Shi, Y., 2000. Comparing inertia weights and constriction factors in

particle swarm optimization. Proc. Cong. On Evolutionary Computation. 84-88.
[5] Afshar, A., O. Bozog Haddad, M.A. Marino and B.J. Adams., 2007. Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation, Journal of the Franklin Institute: 452-462.
[6] Fathian, M., B. Amiri and A. Maroosi., 2007. Application of honey bee mating optimization algorithm on clustering, Applied Mathematics and Computation, 190: 1502-1513.
[7] Niknam, T., 2008. Application of Honey Bee Mating Optimization on Distribution State Estimation Including Distributed Generators, accepted for future publication in Journal of Zhejiang University SCIENCE A.
[8] T. Niknam, J. Olamaie, R. Khorshidi., 2008. A Hybrid Algorithm Based on HBMO and Fuzzy Set for Multi-Objective Distribution Feeder Reconfiguration" World Applied Sciences Journal 4 (2): 308-315,
[9] D. Sedighizadeh, E. Masehian., 2009. Particle Swarm Optimization Methods, Taxonomy and Applications. International Journal of Computer Theory and Engineering, Vol(1), No. 5.
[10] H.D. Chiang. J.C. Wang, G. Darling, "Optimal capacitors placement, replacement and control in Large- scale unbalanced distribution systems: System modeling and a new formulation", IEEE 1990-7803-1883,pp173-179.
[11] S.Sundhararajan , A. Pahwa, "Optimal selection of capacitors for radial distribution system using a genetic algorithm ", IEEE trans. on power system, Vol 9.No.3 ,August 1994.pp 1409-1504
[12] Wei, K., Zhang, T., Shen, X. and Liu, J., (2007), "An Improved Threshold Selection Algorithm Based on Particle Swarm Optimization for Image Segmentation", Proc. IEEE/ICNC.
[13] Zhang, Y.-N., Hu, Q.-N. and Teng, H.-F., (2008), "Active target particle swarm optimization: Research Articles", J. of Concurrency and Computation: Practice & Experience, Vol.20, No.1, pp.29 – 40.
[14] Xie, X.-F., Zhang, W.-J. and Yang, Z.-L., (2002b), "Adaptive Particle Swarm Optimization on Individual Level", Int. Conf. On Signal Processing (ICSP), pp: 1215-1218.
[15] Pant, M., Thangaraj, R. and Abraham, A., (2008), "Particle Swarm Optimization Using Adaptive Mutation", IEEE/DEXA'08, pp. 519-523.
[16] Zeng, J. and Hu, J. and Jie, J., (2006), "Adaptive Particle Swarm Optimization Guided by Acceleration Information", Proc. IEEE/ ICCIAS, Vol.1, pp.351-355.
[17] Pampara, G., Franken, N. and Engelbrecht, A.P., (2005), "Combining particle swarm optimization with angle modulation to solve binary problems," The IEEE Cong. on Evolutionary Comput., Vol. 1, pp. 89-96.
[18] Riget, J. and Vesterstroem, J. S. (2002), "A diversity-guided particle swarms optimizer – the ARPSO". Technical Report No. 2002-02. Dept. of Computer Science, University of Aarhus, EVA Life.
[19] Sedlaczek, K., Eberhard, P., (2006), "Using Augmented Lagrangian Particle Swarm Optimization for Constrained Problems in Engineering", J. of Structural and Multidisciplinary Optimization, Vol. 32, No. 4, pp. 277-286.
[20] Alviar, J.-B., Pena, J., and Hincapie, R., (2007), "Subpopulation best rotation: a modification on PSO", Revista Facultad de Ingenieria. No.40, pp.118-122.
[21] Kennedy, J., and Eberhart, R.C. (1997), "A discrete binary version of the particle swarm algorithm", Int. IEEE Conf. on Systems, Man, and Cyber., Vol.5, pp.4104 – 4108.
[22] Shi, Y., and Krohling, R. A., (2002), "Co-evolutionary particle swarm optimization to solve min-max problems," in Proc. Cong. On Evolutionary Comput., Vol. 2, pp. 1682-1687.
[23] Jarbouia, B., Cheikha, M., Siarryb, P. and Rebaic, A., (2007), "Combinatorial particle swarm optimization (CPSO) for partitioned clustering problem", J. Applied Mathematics and Computation, Vol. 192, Issue 2, 15 pp. 337-345.
[24] Liang, J., J., Qin, A., K., and Baskar, S., (2006) "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of multimodal Functions" IEEE Trans. Evolutionary Computation, Vol. 10, No. 3.
[25] Baskar, S. and Suganthan, P.N., (2004) "A novel concurrent particle swarm optimization," in Proc. IEEE Cong. on Evolut. Comput., Vol. 1, pp. 792-796.
[26] Aguirre, A. H., Munoz Zavala, A. E. and Diharce, E. V., and Botello Rionda, S. (2007), "COPSO: Constraints Optimization via PSO algorithm", Comunicacion Tecnic, (CC/CIMAT).
[27] Niu, B., Zhu, Y., and He, X., (2005), "Multi-population cooperative particle swarm optimization," in Proc. European Conf. on Artificial Life, M. C. et al., pp. 874-883.
[28] Van den Berg, F., and Engelbrecht A.P., (2004), "A cooperative approach to particle swarm optimization", IEEE Trans. On Evolutionary Comput., Vol. 8, No. 3, pp. 225-239

[29] Yao, X., (2008), "Cooperatively Coevolving Particle Swarms for Large Scale Optimization", Conf. of EPSRC, Artificial Intell. Technologies New and Emerging Computer Paradigms.

[30] Felix T. S. Chan1, Kumar, V. and Mishra, N., (2007), "A CMPSO algorithm based approach to solve the multi-plant supply chain problem", Swarm Intell.: Focus on Ant and Particle Swarm Optimization, pp. 532.

[31] Jingbo, A. and Hongfei, T., (2005), "Cultural based Particle Swarm Optimization", Center for Science and Technology Development, Ministry of Education P. R. China.

[32] Xie, X.-F., Zhang, W.-J., and Yang, Z.-L., (2002c), "A Dissipative Particle Swarm Optimization", Cong. on Evolutionary Comput. (CEC), pp.1456-1461.

[33] Ji, C., Zhang, Y., Gao, SH., Yuan, P. and Li, Zh., (2004), "Particle swarm optimization for mobile ad hoc networks clustering", IEEE Int. Conf. on Networking, Sensing and Control, Vol 1, pp. 372 – 375.

[34] Lian, Z., Gu, X., and Jiao, B., (2006), "A Dual Similar Particle Swarm Optimization Algorithm for Job-Shop Scheduling With Penalty," in Proc. IEEEWorld Cong. on Intelligent Control and Automation, Vol. 2, pp. 7312-7316.

[35] Shen, X., Wei, K., Wu, D., Tong, Y. and Li, Y., (2007), "A Dynamic Adaptive Dissipative Particle Swarm Optimization with Mutation Operation", Proc. IEEE/ ICCA, pp. 586-589.

[36] Liao, c.-y., Lee, W.-P., Chen, X., (2007), "Dynamic and adjustable particle swarm optimization, Proc. of the 8th Conf. on 8th WSEAS Int. Conf. on Evolutionary Computing, Vol. 8.

20/5/2013

[37] Cui, Zh. Zeng, J. And Cai, X., (2004), "A guaranteed convergence dynamic double particle swarm optimizer", Fifth World Cong. On Intell., Control and Automation, vol. 3, pp. 2184 – 2188.

[38] Subrarnanyam, V. Srinivasan, D. and Oniganti, R. (2007) "Dual layered PSO Algorithm for evolving an Artificial Neural Network controller", IEEE/CEC, pp. 2350-2357.

[39] Hu, X. and Eberhart, R. C. (2002) "Multi objective optimization using dynamic neighborhood particle swarm optimization". Proc. of the IEEE/ CEC, pp. 1677-1681

[40] Kulkarni, R.V. and Venayagamoorthy, G.K., (2007), "An Estimation of Distribution Improved Particle Swarm Optimization Algorithm", Proc. IEEE/ ISSNIP, pp. 539-544.

[41] Lee, T.-Y., (2007b), "Optimal Spinning Reserve for a Wind-Thermal Power System Using EIPSO", IEEE/ TPWRS, Vol. 22, No.4, pp. 1612– 1621.

[42] Wei, C., He, Z., Zhang, Y., and Pei, W. (2002). "Swarm directions embedded in fast evolutionary programming". In Proc. of the IEEE/CEC, pp. 1278–1283.

[43] Jun-jie, Xu. and Zhan-hong, Xin., (2005) "An extended particle swarm optimizer" Proc. IEEE Symp, Parallel and Distributed.

[44] Poli, R., Langdon,W. B., and Holland, O. (2005b). "Extending particle swarm optimization via genetic programming. In M. Keijzer et al. (Eds.), Lecture notes in computer science: Vol. 3447. Proc. of the 8th European conf. on genetic programming, pp. 291–300, Springer.