Investigation of Hybrid Particle Swarm Optimization Methods for Solving Transient-Stability Constrained Optimal Power Flow Problems

K. Y. Chan, G. T. Y. Pong and K. W. Chan

Abstract—**In this paper, hybrid particle swarm optimization (PSO) is proposed for solving the challenging multi-contingency transient stability constrained optimal power flow (MC-TSCOPF) problem. The objective of this nonlinear optimization problem is to minimize the total fuel cost of the system and at the same time fulfil the transient stability requirements. The optimal power flow (OPF) with transient stability constraints considered is re-formulated as an extended OPF with additional rotor angle inequality constraints, which is suitable for hybrid PSO to solve. Comparison between various existing hybrid PSO techniques is carried out by solving the New England 39-bus system. Experimental results indicate that the hybrid PSO integrated with the mutation operation of genetic algorithms is better than the other existing hybrid PSO methods in both solution quality and stability. As a result, reasonable solutions can be reached with faster convergence speeds and smaller computational efforts.**

Index Terms— **Particle swarm optimization, genetic algorithms, transient stability, optimal power flow, constrained optimization.**

I. Introduction

The MC-TSCOPF aims to achieve an optimal solution of a specific objective function, such as fuel cost, network loss, by setting the system control variables, while satisfying the system to withstand specified contingencies (disturbances) and reach an acceptable steady-state operating condition [1]. In solving MC-TSCOPF, the difficulty mainly comes from the non-convexity nature of OPF and the nonlinear differential-algebraic equations which describe the transient stability constraints of the power system. Nonlinear and semi-infinite programming [2,3] was proposed to solve the MC-TSCOPF. However, not only their formulation is complex and heavily tied to the system models, but also they rely on convexity to obtain the global optimum solution and as such are forced to simplify relationships in order to ensure convexity [4].

Particle swarm optimization (PSO) is a recently proposed population based stochastic optimization algorithm which was inspired by the social behaviors of animals such as fish schooling and bird flocking [5]. Compared with other stochastic optimization methods, PSO has comparable or even superior search performance for many hard optimization problems with faster and stable convergence rates [6]. It requires only few parameters to be tuned and hence is attractive from an implementation viewpoint. It has attracted broad attention in the fields of evolutionary computing, optimization and many others. In recent years there have been a lot of reported works focused on the PSO. It has been applied widely in the function optimization, artificial neural networks development, fuzzy control and some other fields.

However, it can be noticed that PSO performs well in the early state of the search, but the improvement decreases gradually along the searching stages. Its improvement even terminates in the later stages of the search. It behaves like the traditional local searching methods that drop into a local optima and cannot escape from it.

Many improved PSO algorithms have been proposed by incorporating with other optimization methods, so as to explore better solutions. It can be found from the literatures that PSO algorithms could be enhanced by incorporation with GAs [7-10]. In these approaches, operations of PSO and GAs are crossed over the search simultaneously. GAs' operations like crossover, mutation, selection are integrated into PSO. However, so far no conclusive conclusion has been reached in which hybrid PSO algorithm is better than the others.

In this paper, we re-formulated a simple transformation of the multi-contingency-transient stability constraints to the optimal power flow problem, which is suitable for PSO algorithms to solve. A refined PSO is adopted as the main solver for this challenging MC-TSCOPF problem. Among all the existing hybrid PSO algorithms, four selected ones [7-10] have been implemented and tested on the New England 39 bus system. Experimental result shows that the hybrid PSO algorithm which integrates PSO with the mutation operation of GAs is better than the other existing hybrid PSO algorithms in both solution quality and solution stability in which smaller computational effort is required.

II. MC-TSCOPF problem formulation

MC-TSCOPF is mathematically defined as

$$\min \ f(\boldsymbol{x},\boldsymbol{y}) \tag{1}$$

$$s.t. \quad \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y})=0 \tag{2}$$

$$\boldsymbol{H}(\boldsymbol{x},\boldsymbol{y})\le 0 \tag{3}$$

$$\boldsymbol{U}(\boldsymbol{x}(t),\boldsymbol{y})\le 0,\ t\in \boldsymbol{T} \tag{4}$$

The authors gratefully acknowledge the support by the Hong Kong Polytechnic University (Project A-PA4Y and 4-ZV28).

K.Y. Chan is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR. (richanky@inet.polyu.edu.hk)

G.T.Y. Pong is with the Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong SAR. (Glory.Pong@polyu.edu.hk).

K. W. Chan is with the Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong SAR (eekwchan@polyu.edu.hk)

where $\boldsymbol{x}(t)$ is a dependent vector which includes active and reactive power of the swing bus, voltage angle and reactive power of generator buses, and voltage angle and magnitude of load buses. $\boldsymbol{T} = [t_0, t_{cl}) \cup (t_{cl}, t_e]$ is the transient period from the occurrence of the disturbance at time t_0 to the clearing time t_{cl} and then to the ending time t_e. $\boldsymbol{x}$ represents the initial value of $\boldsymbol{x}(t)$ at $t = 0$. $\boldsymbol{y}$ is a control which includes vector active power and voltage magnitude of generator buses, voltage angle and magnitude of the swing bus, and tap position of LTCs. f can be expressed as the total generation cost, total network loss, corridor transfer power, total cost of compensation, etc. $\boldsymbol{g}$ is the set of equality constraints which are usually the power flow constraints for a specified operating condition. $\boldsymbol{H}$ is inequality constraints for the steady-state security limits like bus voltage magnitude limits, generator power limits, thermal limits for transmission lines, etc. The dynamic security constraints set $\boldsymbol{U}$ is infinite in the functional space. Further details of the formulation of MC-TSCOPF are available in [3,11].

Since the equality constraints $\boldsymbol{g}$ are imposed implicitly by the power flow calculation incorporated within the algorithm and also the inequality constraints $\boldsymbol{H}$ is directly satisfied by the PSO, the MC-TSCOPF can be formulated as a penalty function problem:

$$\tilde{F}(\boldsymbol{x}) = \min\left\{ f(\boldsymbol{x}, \boldsymbol{y}) + \beta \max[\boldsymbol{U}(\boldsymbol{x}(\boldsymbol{t}), \boldsymbol{y})^2] \right\} \quad (5)$$

Generally, transient stability constraints can be considered as hard constraints that should not be violated whilst the static constraints are soft in nature that slight violation could be tolerant. Compared with other constraint handling approaches [12,13], penalty function offers a simple and flexible strategy to effectively deal with mixed hard and soft constraints. In addition, there is no need to have separate penalty factors for each type of constraints. In (5), any transient instability would introduce a huge angle deviation and thus produce a large violation and thus discrimination even though the same penalty factor is used for all type of violations. Typically, $\beta = 1000$ works very well in most power systems [11].

III. TSCOPF using Particle Swarm Optimization

PSO is a novel optimization method developed by Kennedy and Eberhart [5,6]. This type of algorithms is modeled on processes of the sociological behaviour associated with bird flocking, and is one of the evolutionary computation techniques essentially. It uses a number of particles that constitute a swarm. Each particle traverses the search space looking for the global minimum (or maximum). In a PSO system, particles fly around in a multidimensional search space. During flight, each particle adjusts its position according to its own experience, and the experience of neighbouring particles, making use of the best position encountered by itself and its neighbours. The swarm direction of a particle is defined by the set of particles neighbouring the particle and its history experience.

The best previous position of a particle is recorded and represented as *pbest*. The position of the best particle among all the particles is represented as *gbest*. The velocity and position of each particle can be calculated using the following formulas [14]:

$$\begin{aligned} v_{i+1} &= k \cdot (w \cdot v_i + \varphi_1 \cdot rand() \cdot (pbest - x_i) \\ &\quad + \varphi_2 \cdot rand() \cdot (gbest - x_i)) \\ x_{i+1} &= x_i + v_{i+1} \end{aligned} \quad (6)$$

where x_i and v_i are the current position and velocity of particle at the i^{th} generation respectively, w is inertia weight factor, φ_1 and φ_2 are acceleration constants, *rand()* returns a uniform random number in the range of [0,1], k is constriction factor derived from the stability analysis of equation (6) to ensure the system to be converged but not prematurely [15]. Mathematically, k is a function of φ_1 and φ_2 as reflected in the following equation:

$$k = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad (7)$$

where $\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$.

PSO utilizes *pbest* and *gbest* to modify the current search point to avoid the particles moving in the same direction, but to converge gradually toward *pbest* and *gbest.* Suitable selection of inertia weight w provides a balance between global and local explorations. Generally, w can be dynamically set with the following equation [6]:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{i_{\max}} \times i \quad (8)$$

where $i_{\max}$ is the maximum number of iterations, i is the current number of iterations, $w_{\max}$ and $w_{\min}$ are the upper and lower limits of the inertia weight, and are set to 1.2 and 0.1, respectively, in the case studies.

In the above procedures, the particle velocity is limited by a maximum value $v^{\max}$. The parameter $v^{\max}$ determines the resolution, or fitness, with which regions are to be searched between the present position and the target position. This limit enhances the local exploration of the problem space and it realistically simulates the incremental changes of human learning. If $v^{\max}$ is too high, particles might fly past good solutions. If $v^{\max}$ is too small, particles may not explore sufficiently beyond local solutions. Based on our experiences with PSO, $v^{\max}$ is often set at 10% to 20% of the dynamic range of the variable on each dimension. In this paper, 20% of the variable dynamic range is adopted as the limit of $v^{\max}$.

The following describes the incorporation of PSO algorithm (PSOTSCOPF) into the multi-contingency transient stability constrained optimal power flow.

Step 1: Input system data, contingency set, PSO parameters and specify the lower and upper boundaries of each variable. Control variables include the active power and terminal voltage of each generator, voltage angle and magnitude of the swing bus, and tap position of each LTC.

Step 2: Each particle in the swarm represents a feasible candidate solution to the optimization problem and is initialized randomly with all control variables satisfied their practical operation constraints.

Step 3: For each particle, an unconstrained Newton-Raphson power flow calculation is used to determine the power flow solution, which includes all the dependent variables, for a given set of control variables.

Step 4: Evaluate the fitness of each particle using the evaluation function described in (5). Power flow solution obtained in Step 3 is used to evaluate the objective function (1) and the static violations (2)-(3). For transient stability violation evaluation (4), transient stability simulation is used to produce the generator rotor responses. The maximum rotor angle deviation from the COI, among all generators and contingencies, is then used to compute a transient stability penalty using (5).

Step 5: Find the best position of the swarm *gbest* and the best position of each particle *pbest* by comparing the evaluation value $\tilde{F}(x)$ of each particle with the one in *pbest*. If $\tilde{F}(x)$ is better, then set *pbest* to the corresponding *x*. The best among *pbest* is denoted as *gbest*.

Step 6: If there is any stopping criteria being satisfied, go to Step 11; otherwise, increment the iteration number *i*.

Step 7: Update the inertia weight *w* according to equation (8).

Step 8: Update the velocity *v* of each particle according to equation (6).
If $v > v^{max}$, $v = v^{max}$. If $v < -v^{max}$, $v = -v^{max}$.

Step 9: Update the position of each particle. If a particle violates its position limits (i.e. limits of the control variables) in any dimension, set its position at the proper limit.

Step 10: Return to Step 4 to repeat the evaluation process with updated position, until the termination condition is reached.

Step 11: The particle that generates the latest *gbest* is the optimal value.

IV. Hybrid Particle Swarm Optimization Method

This section presents the operation of the four selected hybrid PSO methods [7-10] for solving the MS-TSCOPF problem:

A. Ahmed et al's hybrid PSO (AhmedPSO)

Ahmed et al [7] observed that PSO performs well in the early iterations, but it usually presents problems reaching a near-optimal solution. The behavior of the PSO in the model presents some important aspects related with the velocity update. If a particle's current position coincides with the global best position, the particle will only move away from this point if its inertia weigh *w* and velocity v_i are different from zero. If their velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle, which may lead to a premature convergence to the PSO. In fact, this does not even guarantee that the PSO has converged on a local minimum—it merely means that all the particles have converged to the best position discovered so far by the swarm. This phenomenon is known as *stagnation* [15]. To prevent it, Ahmed et al proposed to integrate the mutation of GAs into the PSO. This approach allows the search to escape from local optima and search in different zones of the search space. It starts with the random choice of a particle in the swarm and moves to different positions inside the search area. Ahmed et al employed the mutation operation by the following equation:

$$mut\left(p[k]\right) = p\left([k]\times -1\right) + \omega \tag{9}$$

where $p[k]$ is the random choice particle from the swarm, and ω is randomly generated within the range $\left[0, 0.1\times\left(x_{max} - x_{min}\right)\right]$, representing 0.1 times the length of the search space. The procedures of Ahmed et al [7]'s PSO (AhmedPSO) are shown:

Step 1: Step 1 to Step 9 of PSOTSCOPF

Step 2: Perform the mutation operation based on (9)

Step 3: Step 10 to Step 11 of PSOTSCOPF

B. Juang's hybrid PSO (JuangPSO)

Juang [8] observed that GA and PSO work with a population of solutions. Originally, PSO works based on social adaptation of knowledge, and all particles are considered to be in the same iteration. On the contrary, GA works based on evolution from iteration to iteration, and the changes of particles (or chromosomes in GA's terminology) in a single iteration are not considered.

In the reproduction and crossover operation of GAs, particles are reproduced or selected as parents directly to the next generation without any enhancement. However, in nature, particles will grow up and become more suitable to the environment before producing offspring. To incorporate this phenomenon, PSO that is inspired by social interaction of knowledge is adopted to enhance the top-ranking particles on each iteration. It enhances particles by both sharing information between each other and their individually learned knowledge. Then, these enhanced particles are reproduced and selected as parents for crossover operation and mutation operation as in genetic algorithms. Offspring produced by the enhanced particles are expected to perform better than some of those particles in original iteration, and the poor-performed particles will be weeded out from iteration to iteration. The procedures of Juang's hybrid PSO (JuangPSO) are shown:

Step 1: Step 1 to Step 4 of PSOTSCOPF

Step 2: Select top-half best performing particles as elites.

Step 3: Perform the PSO operation (same as the one shown in Step 5 to Step 9 of PSOTSCOPF) on the selected elites rather than all particles.

Step 4: Select two enhanced elites in Step 3 as two parents by tournament selection, in which two enhanced elites are selected randomly, and their fitness values are compared to select the elite with better fitness values as one parent, and then the other parent is selected in the same way.

Step 5: Produce two offspring by performing two-point crossover on the two parents selected in Step 4.

Step 6: Repeat Step 4 and Step 5 until the reproduced offspring occupies the whole population of the elites.

Step 7: Uniform mutation is adopted on offspring reproduced on Step 6, that the mutated gene is drawn randomly, uniformly from the corresponding search interval. A constant mutation probability $p_m = 0.1$ is used.

Step 8: Replace the bottom-half worst performing particles by the offspring produced in Step 7.

Step 9: Step 10 to Step 11 of PSOTSCOPF.

C. Noel and Jannett's hybrid PSO (NoelPSO)

Noel and Jannett [9] intended to increase the convergence speed of the PSO by integrating the derivative information of gradient into the formulation of the velocity of each particle as in equation (6). The classical gradient descent is assumed as:

$$x_{i+1} = x_i - \eta \nabla C(x_i) \tag{10}$$

where C is a cost function (in our case the cost function is defined as (5)), η is the learning rate, and x_i is the current position of particle at the i^{th} generation.

An updated equation was proposed by combining (6) and (10):

$$\begin{aligned} v_{i+1} &= k \cdot w \cdot v_i + k \cdot \varphi_1 \cdot rand() \cdot (pbest - x_i) \\ &+ \left(\frac{rand()}{\varepsilon}\right) \cdot \left(C(x_i + \varepsilon \cdot E_i) - C(x_i)\right) \\ x_{i+1} &= x_i + v_{i+1} \end{aligned} \tag{11}$$

where w is defined as in (8) and k is defined as in (7), ε is a small constant, E_i is the i^{th} standard basis vector for R^n, and n is the number of variables of the cost function.

The first term in (11) is the inertial term, the second term moves the particle towards the global best solution, and the third term moves the particle in the direction opposite the gradient. Noel and Jannett's hybrid PSO (we call it NoelPSO) is identical to PSOTSCOPF except that the update equation (11) is used in Step 8 instead of using equation (6).

D. Shi's hybrid PSO (ShiPSO)

The main idea of Shi's hybrid PSO [10] is to run PSO and GA methods alternatively in series. It performs a pre-defined number of PSO iterations simultaneously at first. After the PSO iterations, the final particles are constituted the first population of GA. Then the population is evolved using GA-operators until the pre-defined number of iterations of GA reached. After running with the pre-defined number of iterations of GA, the reproduced population of GA is transmitted back to PSO as the first population of particles. Then the PSO operation performs until the pre-defined number of iterations of the second PSO termination condition reached. The procedures of Shi's hybrid PSO are shown:

Step 1: Step 1 to Step 9 of PSOTSCOPF

Step 2: Return to Step 1 to repeat the PSO operations for particles updating until pre-defined number of PSO iterations is reached.

/ Step 1 to Step 2 are the steps of PSO */*

Step 3: Pass the final population of particles of the PSO to GA as its first population.

Step 4: Select the parents in the population based on the roulette-wheel selection.

Step 5: Product offspring by performing discrete crossover on the selected parents with the crossover rate $p_c = 0.8$.

Step 6: Mutate the produced offspring by performing mutation operator of Gaussian perturbation with the mutation rate $p_m = 1/n$, where n is the number of variables of the cost function.

Step 7: Return to Step 4 to repeat the evaluation process until the pre-defined number of iterations of GA is reached.

/ Step 3 to Step 7 are the steps of GA */*

Step 8: Pass the final population of GA to the second PSO as its first particles population.

Step 9: Step 1 to Step 9 of PSOTSCOPF

Step 10: Return to Step 9 to repeat the PSO operations for particles updating until the termination condition is reached.

/ Step 8 to Step 10 are the steps of PSO */*

V. Case Study

A case study of solving the optimal power flow problems with stability constraints on the New England 39-bus system is used to demonstrate the effectiveness and robustness of the hybrid PSO based approaches (PSOTSCOPF, AhmedPSO, JuangPSO, NeolPSO and ShiPSO) for solving MC-TSOCPF problems. All the hybrid PSO based approaches are coded in Matlab. The system data of the power system is collected in [16,17]. The New England 39-bus test system comprises 10-generator, 39-bus, and 46-line. Power System Toolbox [16] is employed to perform time-domain transient stability simulations for determining generator rotor trajectories. The time step adopted is 0.01s and the integration time interval is fixed to 1.5s. The total load for the operating condition considered is 6,098 MW and 1,409 MVAr. There are three onload tap changers connected buses 11-12, 12-13 and 19-20.

After a complete scan of all possible single line fault contingencies, the following two conflicting contingencies were identified.

Contingency 1: A three phase fault occurred at the end of line 26-27 near bus 26. The fault was cleared by tripping the line at bus 26 after 110 ms and at bus 27 after 120 ms.

Contingency 2: A three phase fault occurred at the end of line 16-17 near bus 16. The fault was cleared by tripping the line at bus 16 after 80 ms and at bus 17 after 100 ms.

With the above two contingencies, the following 4 cases were built.

Case 1: conventional OPF without any transient stability constraints

Case 2: transient stability constrained OPF with contingency 1 considered only

Case 3: transient stability constrained OPF with contingency 2 considered only

Case 4: transient stability constrained OPF with contingency 1 and 2 considered

The parameters used in all the hybrid PSO based approaches are followings: swarm size = 30, initial inertia weight w = 1.2, acceleration constants $\varphi_1 = \varphi_2 = 2.05$, penalty factor β = 1000, pre-defined number of iterations = 50. 50 test runs were performed to collect the four statistics for the average, variance, best and worst results among the 50 test runs.

Table I gives the average optimization results of the 50 runs for the above four cases. The number in bracket is their position ranking.

Table I. Mean Cost in 50 Test Runs

Methods	Case 1	Case 2	Case 3	Case 4
PSOTSCOPF	36279 (3)	36557 (4)	36395 (3)	36754 (4)
AhmedPSO	*36240 (1)*	*36405 (1)*	*36306 (1)*	*36629 (1)*
JuangPSO	36288 (4)	36437 (2)	36359 (2)	36666 (2)
NoelPSO	36380 (5)	36703 (5)	36572 (5)	36969 (5)
ShiPSO	36257 (2)	36524 (3)	36469 (4)	36750 (3)

It is observed that AhmedPSO algorithm achieves the best mean cost among the five PSO algorithms. In fact, the AhmedPSO obtains the lowest cost values in all cases.

Table II shows the variance of the 50 runs. The smaller the variance means the closer the values cluster around the mean. Since three out of four of the variances of AhmedPSO are the smallest, it demonstrates that the algorithm is capable to approach and keep searching around the optimal mean closer.

Table II. Variance in 50 Runs

Methods	Case 1	Case 2	Case 3	Case 4
PSOTSCOPF	7574	20027	24449	26107
AhmedPSO	*1731*	*4105*	*8939*	20525
JuangPSO	4702	6287	21091	16117
NoelPSO	25777	4783	40399	245074
ShiPSO	5085	14829	22009	*13893*

Therefore these results indicate that AhmedPSO algorithm is better than the other hybrid PSO based approaches in both solution quality and solution stability in solving the MC-TSOCPF problem. It demonstrates that PSO integrated with the mutation operation of GAs can make enhancement for searching better solutions.

The convergence plots of all the PSO methods for cases 1-4 are shown in Fig 1-4, respectively. They show the progresses of each PSO method through the searches for the first 50 iterations. It can be observed clearly from the figures that the convergence speeds of AhmedPSO with integration of mutation operation are faster than the other four methods whilst its solution is also among the best. In other words, AhmedPSO is more likely to reach better solutions whilst pre-mature convergence is more unlikely to be happened in AhmedPSO than the other four PSO methods.

For accessing the computational efforts required for each hybrid method to reach reasonable solutions, the solutions obtained from the standard PSO, i.e. PSOTSCOPF are used as the acceptable benchmark solutions of the MC-TSCOPF problems. Table III shows the number of iterations needed for each hybrid PSO method to reach the solutions found by PSOTSCOPF with 50 iterations. For the hybrid methods, the maximum number of iterations was set to 50. Any methods which cannot reach an acceptable solution as found by PSOTSCOPF would have a "Nil" in the table. Based on the number of iterations as well as the computation time, as detailed in Table III and IV, needed to reach an acceptable solution, the computational efforts of each hybrid method could be accessed and compared.

It can be found from Table III and IV that AhmedPSO can reach the acceptable solutions with smallest numbers of computational iterations and shortest computational times than the rest four hybrid PSO methods.

Table III. Number of iterations performed in the PSO methods until the acceptable solution reached

Methods	Case 1	Case 2	Case 3	Case 4
PSOTSCOPF	50	50	50	50
AhmedPSO	30	21	24	28
JuangPSO	Nil	26	35	39
NoelPSO	Nil	Nil	Nil	Nil
ShiPSO	45	46	Nil	Nil

Table IV. Computational time (in seconds) performed in the PSO methods until the acceptable solution reached

Methods	Case 1	Case 2	Case 3	Case 4
PSOTSCOPF	13.74	360.54	336.32	680.18
AhmedPSO	8.04	137.82	169.44	371.14
JuangPSO	Nil	187.40	241.66	527.10
NoelPSO	Nil	Nil	Nil	Nil
ShiPSO	15.74	307.48	Nil	Nil

VI. Conclusion

In this paper, four hybrid particle swarm optimization algorithms have been selected from the existing hybrid methods published in recent years for solving the challenging multi-contingency transient stability constrained optimal power flow (MC-TSCOPF) problem. The feasibility and robustness of each hybrid PSO method for the TSCOPF problem are demonstrated on the New England 39-bus system. Experimental results indicate that the superiority of the hybrid PSO method, namely AhmedPSO, which integrate the PSO with mutation operation of GA for solving multi-contingency TSCOPF in both solution quality and stability with smaller computational effort over other existing hybrid PSO methods, have been tested.

Since time-domain simulation is adopted for transient stability evaluation, the computation task of the OPF with transient stability constraints considered is fairly time-consuming. However, the proposed method shows the potential for on-line and off-line applications in a parallel computing environment. This is an area for the future work. Also standard mutation operation with constant mutation space is currently in use, which can be improved by replacing with an enhanced mutation operation with dynamic mutation space. The results will be reported in the near future.

REFERENCES

[1] J. A. Momoh, R. J. Koessler and M. S. Bond: Challenges to Optimal Power Flow, *IEEE Trans. Power Syst.*, vol. 12, no. 1, pp. 444–455, Feb. 1997.

[2] D. Gan, R. J. Thomas and R.D. Zimmerman: Stability-constrained optimal power flow, *IEEE Transactions on Power Systems*, vol. 15, pp. 535-540, 2000.

[3] Y. Xia, K.W. Chan and M. Liu: Direct nonlinear primal-dual interior-point method for transient stability constrained optimal power flow, IEE Proceedings: Generation, Transmission and Distribution, vol. 152, pp. 11-16, 2005.

[4] J. A. Momoh and J. Z. Zhu: Improved interior point method for OPF problems, *IEEE Trans. Power Syst.*, vol. 14, no.3, pp. 1114-1120, Aug. 1999.

[5] J. Kennedy and R. Eberhart: Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp.1942-1948, 1995.

[6] J. Kennedy and R. Eberhart: *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[7] A. A. E. Ahmed, L. T. Germano and Z. C. Antonio: A hybrid particle swarm optimization applied to loss power minimization, IEEE Transactions on Power Systems, Vol. 20, No. 2, pp. 859-866, May 2005.

[8] C. F. Juang: A hybrid genetic algorithm and particle swarm optimization for recurrent network design, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, vol. 34, no. 2, pp. 997-1006, 2004.

[9] M. M. Noel and T. C. Jannett: Simulation of a new hybrid particle swarm optimization algorithm, Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory, pp. 150-153, 2004.

[10] X. H. Shi, Y. H. Lu, C. G. Zhou, H. P. Lee, W. Z. Lin and Y. C. Liang: Hybrid evolutionary algorithms based on PSO and GA, IEEE Congress on Evolutionary Computation, Vol. 4, pp. 2393-2399, 2003.

[11] N. Mo, G. T. Y. Pong, K. W. Chan, S. W. Mei: Multi-contingency Transient Stability Constrained Optimal Power Flow by Genetic Algorithm, Proc. 7th Conf. on Advances in Power System Control, Operation and Management, CDROM, 2006.

[12] R. Farmani and J. A. Wright: Self-Adaptive Fitness Formulation for Constrained Optimization, *IEEE Transactions on Evolutionary Computation*, volume 7, pp. 445-455, 2003.

[13] T. P. Runarsson and X. Yao: Stochastic ranking for constrained evolutionary optimization, *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 284-294, 2000.

[14] B. Zhao, C. X. Guo, and Y. J. Cao: A multiagent-based particle swarm optimization approach for optimal reactive power dispatch, *IEEE Trans. Power Syst.*, vol. 20, no. 2, May 2005.

[15] R. C. Eberhart and Y. Shi: Comparison between genetic algorithms and particle swarm optimization, *Evolutionary Programming VII*. New York: Springer-Verlag, 1998, vol. 1447, Lecture Notes in Computer Science, pp. 611–616.

[16] J. H. Chow: *Power System Toolbox Version 2.0*, Cherry Tree Scientific Software, 2000.

[17] R. Zimmerman and D. Gan: *MATPOWER: A Matlab power system simulation package*, 2005.

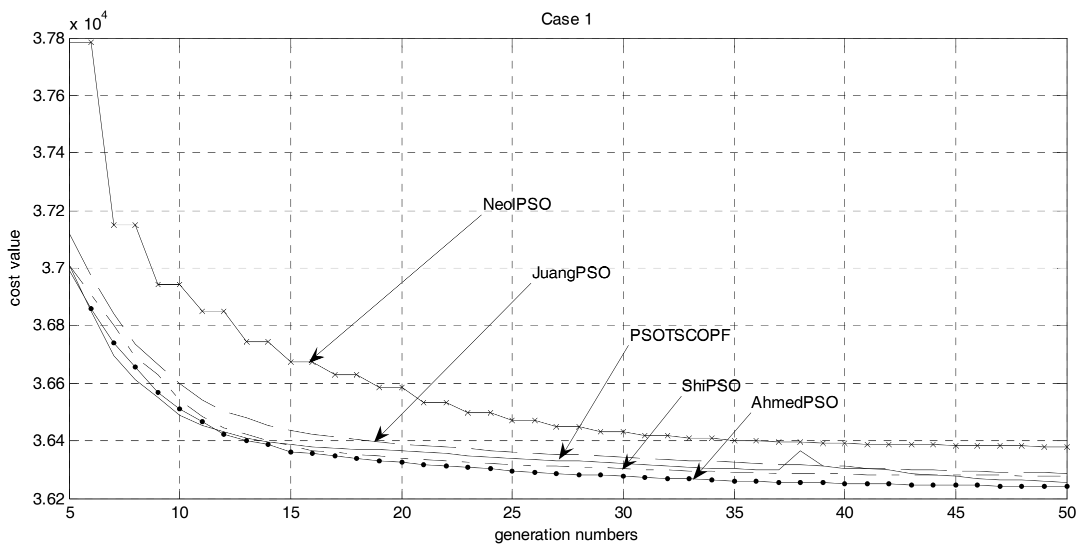

Fig. 1 Convergence curves of various PSO methods for Case 1

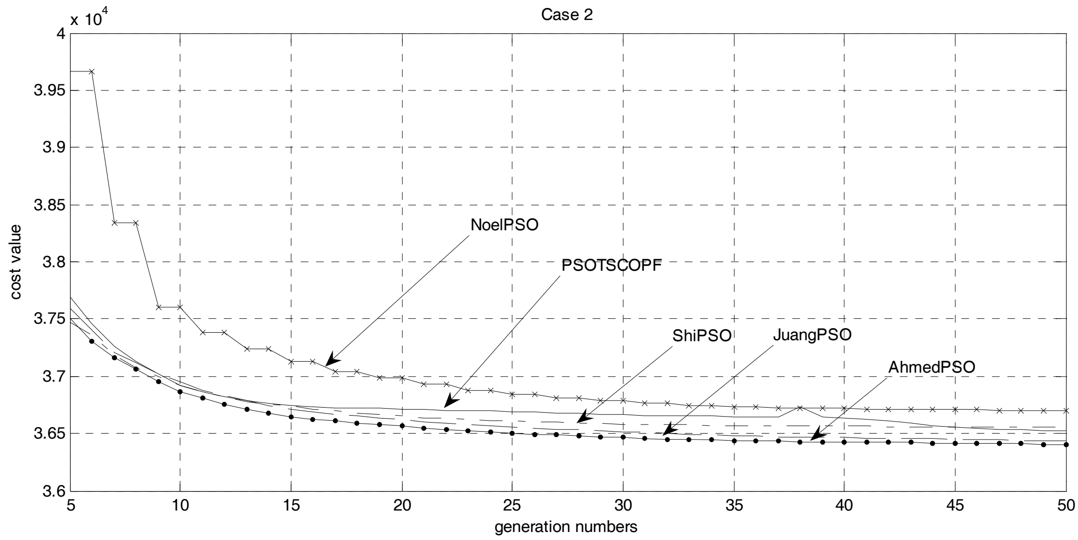

Fig. 2 Convergence curves of various PSO methods for Case 2

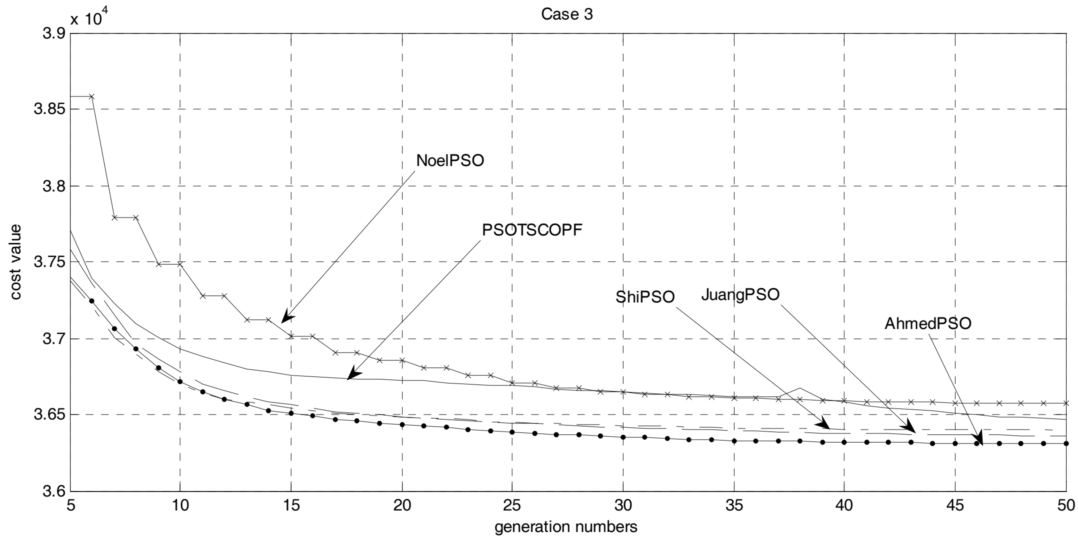

Fig. 3 Convergence curves of various PSO methods for Case 3

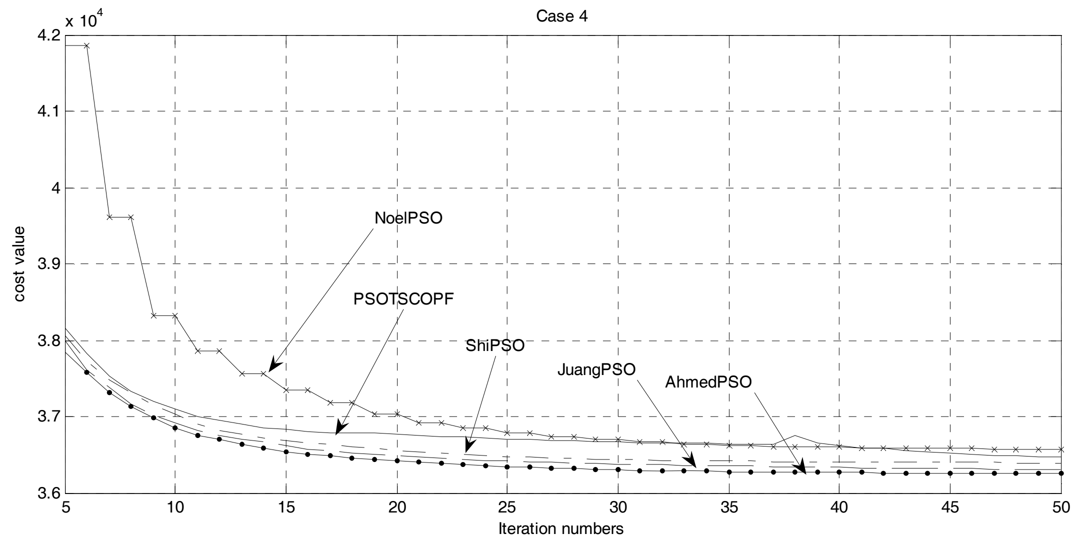

Fig. 4 Convergence curves of various PSO methods for Case 4