$$U_{Gi}^{\min} \le U_{Gi} \le U_{Gi}^{\max} \qquad i = 1,2,...,n_G \qquad (2)$$

b) Susceptance of shunt compensator

$$B_i^{\min} \le B_i \le B_i^{\max} \qquad i = 1,2,...,n_B \qquad (3)$$

c) Transformer tap setting

$$tap_i^{\min} \le tap_i \le tap_i^{\max} \qquad i = 1,2,...,n_T \qquad (4)$$

d) Load bus voltage

$$U_{Li}^{\min} \le U_{Li} \le U_{Li}^{\max} \qquad i = 1,2,...,n_{PQ} \qquad (5)$$

e) Line flow

$$S_{Fi} \le S_{Fi}^{\max} \qquad i = 1,2,...,n_L \qquad (6)$$

where P_{loss} is the total active power losses of the transmission network. The vector of control variables is $\mathbf{u}=[U_G\ B\ tap]^T$and (2)-(4) are the corresponding limits. n_G, n_B and n_T denote the number of generators, shunt compensators and transformers, respectively. The vector of state (dependent) variables is $\mathbf{x}=[U_L\ \ S_F]^T$ and (5)-(6) are the corresponding limits. n_{PQ} and n_L denote the number of load (PQ) buses and transmission lines, respectively. The vector $\mathbf{d}=[P_D\ \ Q_D]^T$ contains active and reactive power demand at all load buses.

In every update of the control vector **u**, the state vector **x**; the load bus voltage both magnitude and angle are computed by the power flow equations given by:

$$0 = P_{Gi} - P_{Di} - U_i \sum_{j=1}^{N} U_j Y_{ij} \cos(\theta_i - \theta_j - \delta_{ij}) \qquad (7)$$

$$0 = Q_{Gi} - Q_{Di} - U_i \sum_{j=1}^{N} U_j Y_{ij} \sin(\theta_i - \theta_j - \delta_{ij}) \qquad (8)$$

where P_{Gi} and Q_{Gi} are active and reactive power generation at bus i, respectively; P_{Di} and Q_{Di} are active and reactive power load at bus i, respectively; Y_{ij} is the admittance matrix corresponding to the ith row jth column and δ_{ij} is the difference in the voltage angle between the ith and jth buses; N is the total number of buses.

III. MEAN-VARIANCE MAPPING OPTIMIZATION ALGORITHM

MVMO operates on a single solution rather than a set of solutions like in many EAs. The internal searching space of all variables in MVMO is restricted in [0,1]. Hence, the real min/max boundaries of variables have to be normalized to 0 and 1. During the iteration it is not possible that any component of the solution vector will violate the corresponding boundaries. To achieve this goal, a special mapping function is developed. The inputs of this function are mean and variance of the best solutions that MVMO has discovered so far. The elegant property of MVMO is the ability to search around the local best-so-far solution with a small chance of being trapped into one of the local optimums. This feature is contributed to the strategy for handling the zero-variance. Further details are referred to our previous work. The procedure of MVMO for solving the ORPD problem with D variables can be summarized as follows.

Step 1: Read power system data and set MVMO parameters

Step 2: **Initialization**: Initialize an initial vector **x** in [0,1] based on the uniform random distribution.

Step 3: **Fitness evaluation**: Denormalize **x,** run the power flow and evaluate the fitness function.

Step 4: **Termination**: Check the termination criteria. If yes, terminate MVMO. Else, **continue** to step 5.

Step 5: **Solution archive**: Store **x**, fitness value, objective value and feasibility to the archive if **x** is better than any of existing solutions.

Step 6: Based on the archived solutions, compute mean $\overline{x_i}$ and variance v_i for each dimension i.

Step 7: **Parent assignment**: Assign the best archived solution $\mathbf{x}_{best}$ as the parent.

Step 8: **Variable selection**: Select $m < D$ dimensions of **x**.

Step 9: **Mutation**: Apply the mapping function to the selected m dimensions.

Step 10: **Crossover**: Set the remaining D-m dimensions of **x** to the values of $\mathbf{x}_{best}$

Step 11: Go to step 3.

Fig. 1 MVMO implementation procedure for ORPD

A. Fitness evaluation and constraint handling

For each individual, a power flow calculation is performed, feasibility of the solution is checked and a fitness value f' is assigned. For a minimization problem, an individual is better if the fitness is smaller. The static penalty scheme is used in this study to handle constraints. Because the control variables in **x** are self-restricted, all dependent variables are constrained by applying the integrated fitness function as follows:

$$\min f' = f + \sum_{i=1}^{n} \lambda_i \max\left[0, g_i\right]^{\beta} \qquad (9)$$

where f is the original objective function; n is the number of constraints; β is the order of the penalty term (usually 1 or 2); λ_i is the penalty coefficient of the constraint i and g_i is the inequality constraint i represented by:

$$g_i(\mathbf{x}, \mathbf{u}) \le 0 \qquad (10)$$

where **x** is the vector of state variables and **u** is the vector control variables. Other constraint handling techniques are also applicable to MVMO.

B. Termination criteria

The MVMO search process is terminated based on completion of the pre-specified number of fitness evaluations (power flow calculations in ORPD) or no improvement in the best fitness over the last s fitness evaluations. In this paper, the first criterion was adopted.

C. Solution archive

The solution archive (SA) serves as the knowledge base of the algorithm for guiding the searching direction. The best n individuals that MVMO has found so far are saved in the SA. Here the data structure of SA is slightly modified from the original one in [15] in order to handle constraints. Two information; fitness and feasibility of each individual are additionally stored as shown in Fig. 2.

#	Objective	Fitness	Feasibility	Solution			
				x_1	x_2	… x_i…	x_D
1							
2							
…							
n							
$\bar{x}_i$	---	---	---				
v_i	---	---	---				

Fig. 2 Data structure of the solution archive

To avoid losing a good solution in constrained optimization problems, the following rules are set up to compare the individual generated at each iteration and existing archived solutions.

- Any feasible solution is preferred to any infeasible solution;
- Between two feasible solutions, the one having better objective value is preferred;
- Between two infeasible solutions, the one having smaller fitness value (smaller constraint violation) is preferred.

An update takes place only if the new individual is better than those in the archive. The archive size is fixed for the entire process. The archived individuals are dynamically sorted so that the first ranked individual is always the best. Feasible solutions are placed in the upper part of the archive. Among these solutions, they are sorted based on their original objective values. Infeasible solutions are sorted according to their fitness values and then placed on the lower part of the archive. Once the archive is filled up by n feasible solutions, any infeasible candidate solution does not have chance to be saved in the archive.

D. Reproduction

Reproduction refers to the process in which an offspring is created. In MVMO, it refers to the steps 7-11 of the procedure shown in Fig. 1. The details of each component are described in the following subsections.

1) Parent assignment

The first ranked (best-so-far) solution denoted as $\mathbf{x}_{\text{best}}$ is assigned as the parent.

2) Variable selection

MVMO searches around the mean saved in the archive for the better solution only in m selected directions. This means that only these m selected dimensions of the offspring will be updated while the remaining D-m dimensions take the corresponding values from $\mathbf{x}_{\text{best}}$. Four strategies for selecting the variables were implemented in MVMO as shown in Fig. 3. From our experience, strategies 2 to 4 generally perform better than strategy 1. However, this observation is still neither general nor conclusive. The performance is also problem-dependent.

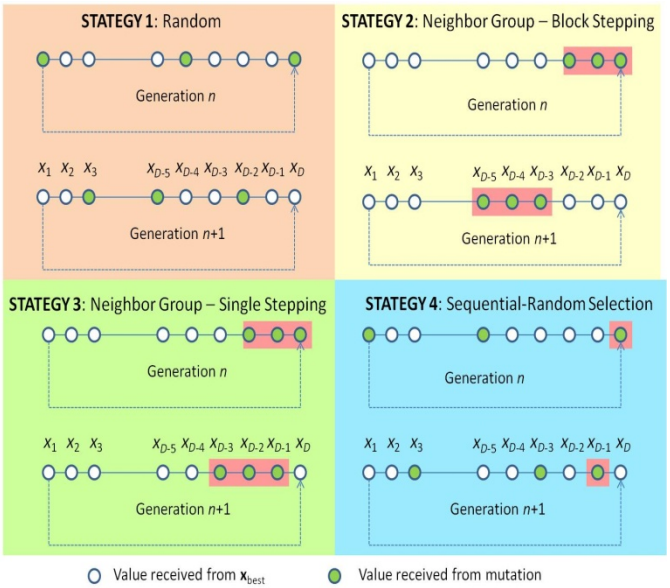

Fig. 3 Variable selection strategies (m=3)

3) Mutation

For each of the m selected dimension, mutation is used to assign a new value of that variable. Given a uniform random number $x_i^{'} \in [0,1]$, the new value of the i component x_i is determined by:

$$x_i = h_x + (1 - h_1 + h_0) \cdot x_i^{'} - h_0 \tag{11}$$

where h_x, h_1 and h_0 are the outputs of the transformation mapping function based on different inputs given by:

$$h_x = h(u_i = x_i^{'}), \quad h_0 = h(u_i = 0), \quad h_1 = h(u_i = 1) \tag{12}$$

Note that the output of (11) is always inside [0,1] for every generated x_i. The mapping function is parameterized as follows:

$$h(\bar{x}_i, s_{i1}, s_{i2}, u_i) = \bar{x}_i \cdot (1 - e^{-u_i \cdot s_{i1}}) + (1 - \bar{x}_i) \cdot e^{-(1-u_i) \cdot s_{i2}} \tag{13}$$

where s_{i1} and s_{i2} are shape factors allowing asymmetrical slopes of the mapping function. Interested readers are referred to our previous work [15] for further details on how to set the two different shape factors. The slope is calculated by:

$$s_i = -ln(v_i) \cdot f_s \tag{14}$$

where f_s is a MVMO parameter namely the shape scaling factor.

Fig. 4 shows an example of the mapping function with a uniform shape factor of 10 for two optimization variables x_1 and x_2. The actual range of the two variables is -2.048 ≤x_1,x_2≤ 2.048. Observe that the mean $\bar{x}_i$ is an input of (13). Assume that the mean of the two variables is at -1 and 1, respectively. In the normalized scale, these correspond to 0.256 and 0.744. It is clearly shown in the left of Fig. 4 that the curves are flat

around the two mean values. This means that most of the samples drawn from these functions will center around -1 and 1 in the actual scale. However, notice that the range of these functions span over the entire range between 0 and 1. This characteristic gives chances for random samplings to cover the entire search space by getting away of the mean values. The right of Fig. 4 shows 10000 samples randomly generated from the mapping function in the left.

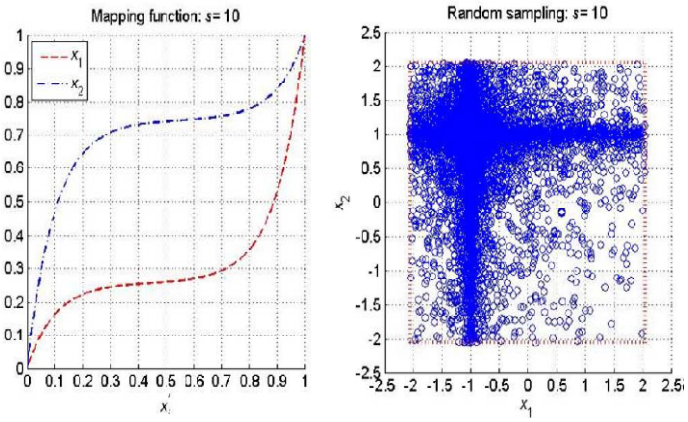

Fig. 4 Mapping function (left) and corresponding two dimensional searching space (right)

4) Crossover

For the remaining *D-m* dimensions, the values of $\mathbf{x}_{best}$ are inherited. This process is equivalent to evolutionary processes where an offspring is a combination of two parents.

IV. SIMULATION RESULTS

The IEEE 57- and 118- bus systems are used as the test cases and named as case 57 and case 118, respectively to examine the performance of MVMO and compare it with other heuristic algorithms. The system data of the two cases are taken from [17]. For both test systems, the lower and upper limits of load bus voltages are 0.95 p.u. and 1.05 p.u., respectively. Generator voltages at the high voltage terminal are defined as continuous variables. The lower and upper limits are set to 0.94 p.u. and 1.06 p.u., respectively. Discrete control variables consist of transformer tap positions and the susceptance of shut compensators. All under-load tap changing (ULTC) transformers are assumed to have 21 discrete taps within ±10% of the nominal voltage (1% for each tap). Each transformer tap is defined by an integer between -10 to 10. These ULTC data are fictitious values. The number of taps and the voltage range in practical cases can be different. All shunt compensators have 11 discrete steps of different ratings (defined by an integer between 0 to 10).

For the two test cases, the performance of MVMO is compared with the following algorithms.

1. PSO: a standard PSO version 2007 [18];
2. DE: a basic DE namely "DE/current-to-best/1" [19, 20],
3. JADE: an adaptive DE algorithm [13] and
4. JADE-vPS: a modified JADE algorithm [21].

The difference between the latter two algorithms is the adaptive scheme for the population size in JADE-vPS. In PSO, the learning rate and inertia weight are internally specified according Clerc's stagnation analysis. The crossover rate *CR* and the scaling factor *F* of DE are specified by 0.2 and 0.6, respectively. The parameters *CR* and *F* are self-adapted in JADE whereby the population size *PS* must be specified by the user. In JADE-vPS, the parameter *PS* is dynamically adjusted during the searching process. The internal parameters of MVMO are set as follows: $f_s = 1$; $AF = 2.5$ and $s_d = 25$. The variable selection is selected to the strategy 3 (see Fig. 3) and the number of random variable *m* is chosen to 6 at the beginning and progressively declined to 2 in every 1000 FEs. The size of solution archive is fixed to 2.

The termination criterion is the maximum number of function (power flow) evaluations (FEs). All programs were implemented in MATLAB R2010a. The computing environment is Intel Core 2 Quad CPU 2.83 GHz and 3.25 GB RAM memory. The software package namely PAST is used. Because the most time-consuming parts in these methods are the repeated power flow calculations and the number of such calculations is fixed, the computational time of all algorithms is not significantly different. The comparison in this paper will be based on quality of the final results.

A. Simulation results- case 57

The IEEE 57-bus system consists of seven generators, 80 lines where 15 of which are equipped with ULTC transformers. Shunt reactive power compensators are connected to buses 18, 25 and 53. The limit of these susceptances is [0,0.2], [0,0.18] and [0,0.18], respectively. Therefore, the ORPD search space has 25 dimensions. The population size *PS* of PSO, DE and JADE and the initial value of *PS* in JADE-vPS is set to 50.

Table I Statistical results – case 57

	MVMO	PSO	DE	JADE	JADE-vPS
Minimum	24.8512	24.8479	24.8360	24.8493	24.8451
Average	24.9917	24.9336	24.8701	24.9494	24.9565
Maximum	25.2608	25.1642	25.0307	25.2044	25.3768
Standard deviation	0.1029	0.0671	0.0352	0.0666	0.0896

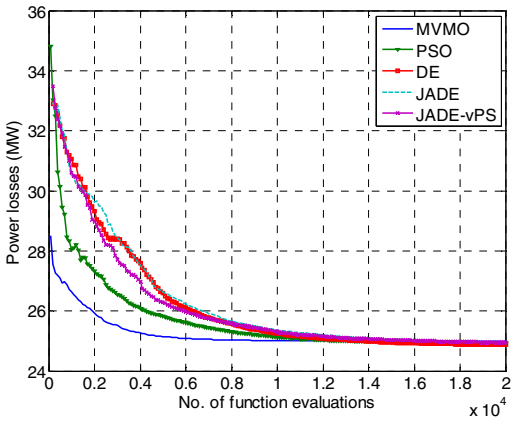

Fig. 5 Average convergence characteristics - case 57

To fairly compare the proposed MVMO method with others, every algorithm is independently run for 50 times. Then statistical values consisting of minimum, average, maximum and standard deviation of active power losses are computed as listed in Table I. The average convergence of active power losses found by each algorithm is plotted after the first 100 FEs as shown in Fig. 5. It is clearly shown that the convergence of MVMO is the fastest. In this test case, the statistical results of MVMO in Table I are not outstanding the other algorithms. However, the MVMO results are on average very close to the other techniques. An interesting observation made from Fig. 5 is that MVMO is very fast in the global search capability because the lowest power loss has been found after the first 100 FEs.

As mentioned in [22] that there are five buses (buses 25, 30, 31, 32 and 33) in this network that the voltages are outside the limits. After the ORPD result given by each method, power flow is calculated to determine bus voltages as shown in Fig. 6. It is shown that all bus voltages can be maintained within the limits. These voltage profiles confirm the merits of ORPD in achieving both reduced power losses and voltage security.

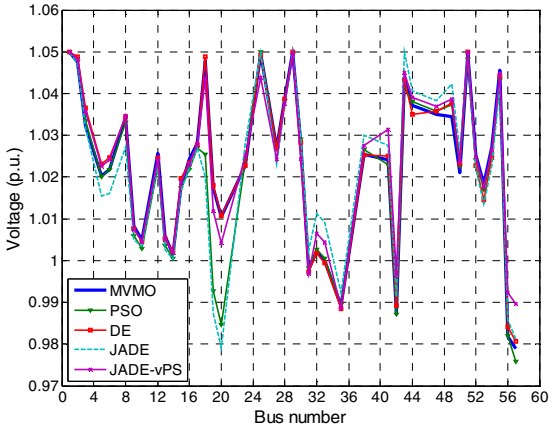

Fig. 6 Load bus voltage profiles – case 57

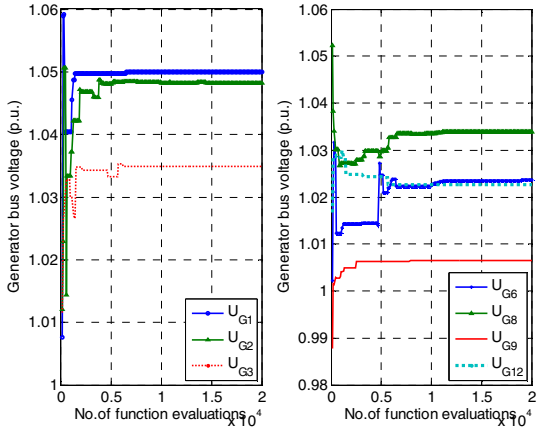

Fig. 7 Convergence of generator voltages –case 57

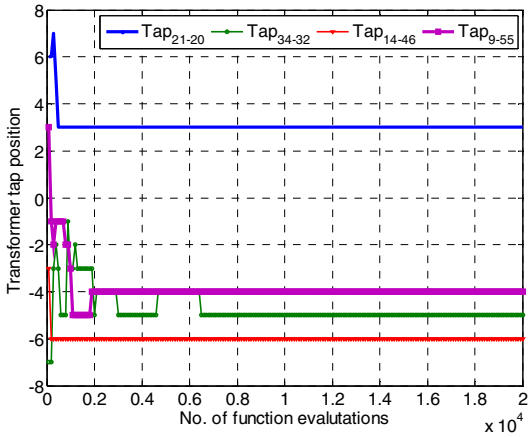

Fig. 8 Convergence of selected transformer taps –case 57

The convergence of optimized control variables are shown in Fig. 7-Fig. 9. From these figures, the control variables change abruptly at the early searching stage. Then, they settle to a steady state at the later stage. At this phase, an optimum has been discovered. The CPU time of all methods is approximately 5 minutes.

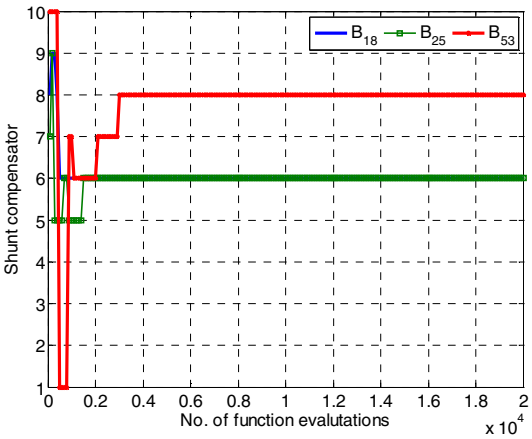

Fig. 9 Convergence of shunt compensator –case 57

B. Simulation results- case 118

To further evaluate the performance of MVMO, the IEEE 118-bus system is employed. In this test case, the search space of this problem has 77 dimensions. The continuous variables consist of 54 generator bus voltages. There are 23 discrete variables; 9 transformer taps and 14 shunt compensators. All shunt compensators are defined by susceptances with the lower limit of 0 p.u. and the upper limit of 0.2 p.u. The population size *PS* of PSO, DE and JADE and the initial value of *PS* in JADE-vPS is set to 100.

The convergence of active power losses averaged from 50 independent trials of different algorithms is shown in Fig. 10. Similar to the case 57, the first plotting starts from 100 FEs. Statistical results are shown in Table II. In this test case, minimum, average, maximum and standard deviation of power losses from MVMO are the lowest among all methods. In other words, MVMO perform the best in this problem. In

terms of the convergence characteristic, MVMO is the fastest among all methods. After the first 100 FEs, MVMO is able to locate the optimal area in the search space. Moreover, it can continue exploring that area for any better solution without being trapped in one of local optimums. In contrast to other greedy search algorithm such as "DE/best/1" in DE, MVMO overcomes the well-know premature convergence and local stagnation. In other words, MVMO is an enhanced greedy search algorithm.

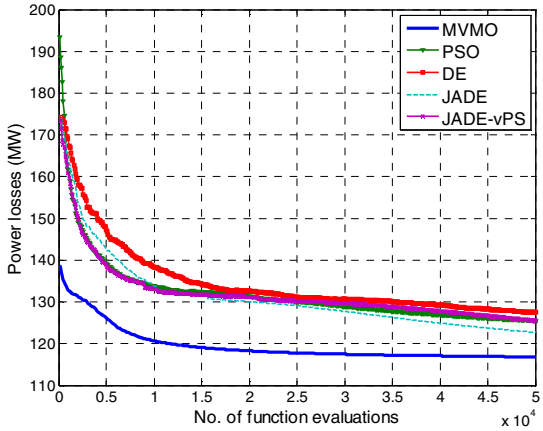

Fig. 10 Average convergence characteristics - case 118

Table II Statistical results – case 118

	MVMO	PSO	DE	JADE	JADE-vPS
Minimum	115.7932	120.8967	125.0250	119.1614	119.2006
Average	116.8202	125.4868	127.4415	122.6774	125.5745
Maximum	119.3584	135.3720	130.2292	132.0349	133.3443
Standard deviation	0.7682	3.1634	1.1337	3.0467	3.8943

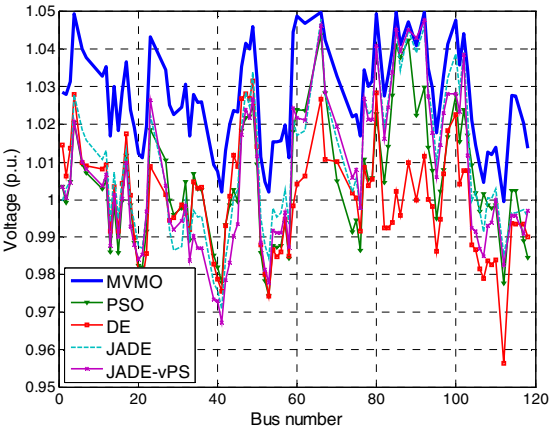

Fig. 11 Load bus voltage profiles – case 118

Fig. 11 shows the voltage profiles at load buses resulting from all methods. Again, all optimization algorithms can maintain all bus voltages within the limits. The CPU time of all methods is approximately 15 minutes.

V. CONCLUSION

MVMO is a novel heuristic algorithm that has recently shown great promises in dealing real-world complex optimization problems. Besides its capability, the algorithm is also simple to be implemented. In this paper, MVMO was applied to solve the optimal reactive power dispatch problem. The performance of MVMO was examined and compared with other heuristic algorithms. The IEEE 57-bus and 118-bus systems were used as the test cases. The simulation results show that MVMO obviously has the better convergence speed than the other algorithms. The final result is nearly the same for all algorithms. . MVMO performs outstandingly in the 118-bus case both in terms of convergence and the minimum reached. . The CPU times used in both cases are relatively high. The source of this occurrence is the power flow program used in this paper. This issue will be corrected in our future work. In summary, MVMO has a huge potential in dealing with large-scale OPF problems.

REFERENCES

[1] H. W. Dommel and W. F. Tinney, "Optimal Power Flow Solutions," *IEEE Trans. Power App. Syst.,* vol. PAS-87, no. 10, pp. 1866-1876, Oct. 1968.

[2] J. A. Momoh, M. E. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993 part I & II," *IEEE Trans. Power Syst.,* vol. 14, no. 1, pp. 96-111, Feb. 1999.

[3] J. L. M. Ramos, A. G. Exposito, and V. H. Quintana, "Transmission power loss reduction by interior-point methods: implementation issues and practical experience," *IEE Proc. Gener. Trans. Distrib.,* vol. 152, no. 1, pp. 90-98, Jan. 2005.

[4] V. H. Quintana and M. Santos-Nieto, "Reactive power-dispatch by sucessive quadratic programming," *IEEE Trans. Energy Convers.,* vol. 4, no. 3, pp. 425-435, 1989.

[5] M. Todorovski and D. Rajicic, "An initialization procedure in solving optimal power flow by genetic algorithm," *IEEE Trans. Power Syst.,* vol. 21, no. 2, pp. 480-487, May 2006.

[6] Q. H. Wu, Y. J. Cao, and J. Y. Wen, "Optimal reactive power dispatch using an adaptive genetic algorithms," *Int. J. Electr. Power Energy Syst.,* vol. 20, no. 8, pp. 563-569, 1998.

[7] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takamura, and Y. Nakanishi, "A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment," *IEEE Trans. Power Syst.,* vol. 15, no. 4, pp. 1232-1239, Nov. 2000.

[8] A. A. A. Esmin, G. Lambert-Torres, and A. C. Z. d. Souza, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Trans. Power Syst.,* vol. 20, no. 2, pp. 859-866, May 2005.

[9] C. H. Liang, C. Y. Chung, K. P. Wong, X. Z. Duan, and C. T. Tse, "Study of differential evolution for optimal reactive power flow," *IEE Proc. Gener. Trans. Distrib.,* vol. 1, no. 2, pp. 253-260, 2007.

[10] M. Varadarajan and K. S. Swarup, "Network loss minimization with voltage security using differential evolution," *Elec. Power Syst. Research,* vol. 78, pp. 815-823, 2008.

[11] Q. H. Wu and J. T. Ma, "Power system optimal reactive power dispatch using evolutionary programming," *IEEE Trans. Power Syst.,* vol. 10, no. 3, pp. 1243-1249, 1995.

[12] M. Tripathy and S. Mishra, "Bacteria Foraging-Based Solution to Optimize Both Real Power Loss and Voltage Stability