

Swarm Intelligence and Evolutionary Approaches for Reactive Power and Voltage Control

L. Grant, *Student Member, IEEE*, G. K. Venayagamoorthy, *Senior Member, IEEE*, G. Krost, *Member IEEE*
and G. A. Bakare

Abstract – This paper presents a comparison of swarm intelligence and evolutionary techniques based approaches for minimization of system losses and improvement of voltage profiles in a power network. Efficient distribution of reactive power in an electric network can be achieved by adjusting the excitation on generators, the on-load tap changer positions of transformers, and proper switching of discrete portions of inductors or capacitors. This is a mixed integer non-linear optimization problem where metaheuristics techniques have proven suitable for providing optimal solutions. Four algorithms explored in this paper include differential evolution (DE), particle swarm optimization (PSO), a hybrid combination of DE and PSO, and a mutated PSO (MPSO) algorithm. The effectiveness of these algorithms is evaluated based on their solution quality and convergence characteristic. Simulation studies on the Nigerian power system show that a PSO based solution is more effective than a DE approach in reducing real power losses while keeping the voltage profiles within acceptable limits. The results also show that MPSO allows for further reduction of the real power losses while maintaining a satisfactory voltage profile.

I. INTRODUCTION

In order to achieve power system stability it is necessary to facilitate reactive power and voltage control of the power system to keep network parameters within predefined limits. Changes in network topology and loading conditions often cause voltage variations in today's power systems. The reactive power dispatch problem must improve system voltage profiles while minimizing system losses at all times [1, 2]. Reactive power flow can be controlled by adjusting the following:

- On-load tap changers of transformers;
- Generating units' reactive power capability;
- Switched capacitors and inductors;
- Static Var Compensators (SVC);
- Flexible AC Transmission System (FACTS) devices and
- Switching of transmission lines.

This work was supported in part by the National Science Foundation (NSF) CAREER Grant ECCS # 0348221 and the US Department of Education GAANN Fellowship Program.

L. Grant is with the Real-Time Power and Intelligent Systems Laboratory, Missouri University of Science & Technology, MO 65409-0249, U.S.A. (email: lisagrant@ieee.org).

G. K. Venayagamoorthy is with the Real-Time Power and Intelligent Systems Laboratory, Missouri University of Science & Technology, MO 65409-0249, U.S.A. (e-mail: gkumar@ieee.org).

G. Krost is with the Electrical Power Systems Institute, University of Duisburg-Essen, Duisburg Germany (e-mail: gerhard.krost@uni-due.de).

G. A. Bakare is with the Electrical and Electronics Engineering Programme, Abubakar Tafawa Balewa University, PMB 0248, Bauchi, Nigeria (e-mail: bakare_03@yahoo.com).

The control devices have lower and upper limits, making the reactive power and voltage control problem very complex for a large power system utilizing several control devices. Since some controls are continuously adjusted while others have multiple discrete steps, there exist many optimal solutions; therefore, an optimization technique is needed to determine the global optimum solution of the overall reactive power dispatch problem.

Many classical techniques have been studied for use in obtaining optimal power flow [3], [4]. These techniques include nonlinear programming (NLP), mixed integer programming, Newton, and quadratic techniques. The limitations of these methods have been reported in [5]. In response to the deficiencies of the conventional methods, several search techniques have been proposed to eliminate the computational complexity of this problem. The proposed techniques include: expert system (ES), genetic algorithm (GA), tabu search, simulated annealing (SA), particle swarm optimization (PSO) and many others [5] – [11].

In previous work, differential evolution (DE) and PSO were compared on their ability to remove voltage limit violations and reduce power losses on the Nigerian grid system [2]. Both algorithms were shown to be suitable in removing limit violations and PSO was shown to have a higher power loss reduction in some cases as compared to DE. In this paper, the PSO algorithm is combined with an evolutionary concept to enhance its performance [12] – [15]. A mutation operator is introduced into the PSO algorithm and results with the mutated PSO (MPSO) are compared against results with DE and a hybrid algorithm of PSO and DE, known as DEPSO on the Nigerian power system. The results are averaged over a large number of runs to evaluate the effectiveness and the overall computational efficiency of the algorithms. Generators, on-load tap changer positions of transformers and shunt inductors are considered as reactive power control devices like in some of the authors' previous studies [2].

II. PROBLEM FORMULATION

In order to solve optimal reactive power dispatch and voltage control problem, a mathematical model is formulated as follows [1].

$$\text{Min } P_{\text{loss}}(X,U) = \sum_{j=1}^{nl} P_j \quad (1)$$

which is subject to the following load flow equations and constraints [16]:

$$\begin{aligned} G(X,U) &= 0 \\ H(X,U) &\geq 0 \end{aligned} \quad (2)$$

$$\begin{aligned} X^{\min} &\leq X \leq X^{\max} \\ U^{\min} &\leq U \leq U^{\max} \end{aligned} \quad (3)$$

Where P_j is the real power losses in line j and nl is the number of transmission lines and

$$\begin{aligned} X^T &= [V_{L1}, V_{L2} \dots V_{Lnd}, Q_{g1}, Q_{g2} \dots Q_{gng}] \\ U^T &= [V_{g1}, V_{g2} \dots, T_1, T_2 \dots T_{nt}, Q_{c1}, Q_{c2} \dots Q_{Cnc}] \end{aligned} \quad (4)$$

The vector X contains the dependent variables, including load bus voltages V_L and generator reactive power outputs Q_g . The vector U contains the control variables, including the generator voltages V_g , transformer tap settings T , and shunt Var compensation Q_c . The load flow equations, $G(X,U)=0$ and $H(X,U)\geq 0$, are solved using the Newton Raphson load flow with the proposed algorithms to optimize the process [16]. There are also constraints (power flow, generation, switchable VAR source, and security) on the system [2].

III. SWARM INTELLIGENCE AND EVOLUTIONARY APPROACHES

A. Differential Evolution

Differential evolution is a heuristic optimization method developed by Storn and Price in 1995 [17]. DE is used to minimize nonlinear and non-differentiable continuous space functions using floating point numbers to encode the parameter variables. DE can also handle mixed integer discrete continuous optimization problems [18]. DE consists of an initial randomly-generated population that is improved through generations of selection, reproduction, crossover, and mutation until problem convergence is met.

An initial population composed of vectors $U_i^0, i=1,2 \dots np$, is randomly generated within the parameter space. The mutation increments are automatically scaled to the correct magnitude. A tournament selection is used for reproduction where the offspring vectors compete against one of their parents. The parallel version of DE maintains two arrays, each holds a population of np , D -dimensional, real value vectors. The primary array holds the current population vector, while the secondary array accumulates the vectors that are selected for the next generation. In each generation, np competitions are held to determine the composition of the next generation. In the selection process, every pair of randomly chosen vectors U_1 and U_2 defines a vector differential: (U_1-U_2) . Their weighed differential is used to perturb another randomly chosen vector U_3 according to (5).

$$U_3 = U_3 + F \cdot (U_1 - U_2) \quad (5)$$

Where, F is the scaling factor for mutation and its value is typically $(0 \leq F \leq 1.2)$. It controls the speed and robustness of the search; a lower value increases the rate of convergence but also has the risk of becoming stuck at a local optimum. The crossover is a complementary process for DE aiming at reinforcing the prior success by generating the offspring vectors out of the object vectors. In every generation, each primary array vector U_i , is targeted for crossover with a vector like U_3' to produce a trial vector U_i according to (6).

$$U_i = \begin{cases} U_3', & \text{if } rand < CR \\ U_i, & \text{otherwise} \end{cases} \quad (6)$$

Where, CR is a crossover constant and its value is typically in the range $(0 \leq CR \leq 1.0)$. The newly created vector will be evaluated by the objective function and the corresponding value is compared with the target vector. The best fit vector is kept for the next generation as given by (7). The best parameter vector is evaluated for every generation in order to track the progress made throughout the minimization process; thus making the DE elitist method:

$$U_i(t+1) = \begin{cases} U_i(t), & \text{if } fit(U_i(t)) \leq fit(offspring(t)) \\ offspring(t), & \text{otherwise} \end{cases} \quad (7)$$

B. Particle Swarm Optimization

Particle swarm optimization is a population based stochastic optimization technique developed by James Kennedy and Russell Eberhart in 1995 [7], [19], [20]. The PSO algorithm is based on the social interactions of flocks of birds and schools of fish, and has been found to be very robust in solving non-linear problems where multiple optima and high dimensionality exists.

PSO differs from other evolutionary algorithms in that better solutions are evolved through the social interactions of individual particles within the group or swarm. The particles are flown thorough the problem space, and over time converge upon the optimal solution, unlike in genetic algorithms where the weakest individuals are discarded and replaced by each subsequent generation. Each particle in the search space has a dynamically adjustable velocity which changes based on its own experience and the information obtained from other members in the swarm. Each particle stores in memory the coordinates of the problem space associated with the best solution it has found so far or its $pbest$ along with the overall best solution found by the entire swarm or $gbest$ value. Essentially the particle is drawn towards the $pbest$ and $gbest$ values as it moves through the problem space. The velocity and position update equations for the PSO algorithm are provided in (8) and (9).

$$V_{id}(k+1) = w(k) \cdot V_{id}(k) + c_1 \cdot rand_1 \cdot (pbest_{id} - X_{id}(k)) + c_2 \cdot rand_2 \cdot (gbest - X_{id}(k)) \quad (8)$$

$$X_{id}(k+1) = X_{id}(k) + V_{id}(k+1) \quad (9)$$

Where,

$rand_1, rand_2$: uniformly random numbers between 0 and 1.

$V_{id}(k)$: current velocity of individual i in dimension d at iteration k .

$V_{id}(k+1)$: velocity of individual i in dimension d at iteration $k+1$.

$X_{id}(k)$: current position of individual i in dimension d at iteration k .

$X_{id}(k+1)$: position of individual i in dimension d at iteration $k+1$.

$pbest_{id}$: dimension d of the $pbest$ of individual i .

$gbest_d$: dimension d of the $gbest$ of the swarm.

c_1 and c_2 : the weighting of the stochastic acceleration that pull each particle towards $pbest$ and $gbest$ (cognitive and social acceleration constant, respectively).

$w(k)$: inertia weight factor that controls the exploitation and exploration of the search space by dynamically adjusting the velocity and it is computed using (10).

$$w(k) = w^{\max} - \frac{w^{\max} - w^{\min}}{iter^{\max}} \cdot iter \quad (10)$$

$iter^{\max}$: maximum number of iterations.

$iter$: current iteration number.

w^{\max} : maximum inertia weight.

w^{\min} : minimum inertia weight.

The particle velocity is limited by the maximum value v^{\max} . Thus, the resolution and fitness of the search depend on v^{\max} . If v^{\max} is too high, then the particles will move in larger steps and the solution reached may not be optimal. If v^{\max} is too low, then the particles will take a long time to reach the desired solution or even get captured in a local minimum. The maximum velocity is characterized by the range of the i^{th} parameter and is given by (11).

$$v_i^{\max} = \frac{U_i^{\max} - U_i^{\min}}{N} \quad (11)$$

Where, N is a chosen number of intervals in the i^{th} parameter.

C. Hybrid DEPSO

DE and PSO can be combined to create a hybrid algorithm called DEPSO [21]. For the DEPSO method, first the PSO equations (8) and (9) are used to update each particle's solution vector. A mutation is then carried out on each particle using the DE operations using (5), (6), and (7). Therefore, an offspring is created for each PSO particle using DE reproduction and crossover operations described above. Next, each DE offspring competes with its PSO parent particle for placement in the next generation. After the new generation vector of particles is updated, the PSO process is repeated, followed by another DE mutation until the solution convergence criterion is met.

D. Mutated Particle Swarm Optimization (MPSO)

To further improve the diversity in the standard PSO, a mutation operator commonly used in GA [22], [23] is introduced into standard PSO algorithm described above. This mutation increases the diversity of the population by preventing the particles from prematurely converging on a local optimum [24]. In the proposed MPSO approach, standard PSO is used for the first 75 iterations and then the mutation operator is activated for the subsequent iterations until convergence is met. The delay in applying mutation is used because the PSO algorithm is known to converge quickly in the first few iterations and then fitness stalls for a long time before an improvement is achieved. Exploration within 75 iterations was found by a number of experiments to have the best results for the MPSO algorithm for this problem. After each PSO particle's position and velocity are updated using (8) and (9), a mutation is applied to the individual particles' positions, X_{id} , which are chosen using a random number less than a predefined mutation rate of ($0 < \text{mutation rate} < 0.3$). The mutated particle's new position is given in (12). $Randn$ is a random number from a normal distribution.

$$X_{id} = gbest_d + 0.5 \times randn \times gbest_d \quad (12)$$

IV. REALIZATION OF SWARM INTELLIGENCE AND EVOLUTIONARY APPROACHES

The swarm intelligence and evolutionary algorithms compared in this paper for the optimal reactive power dispatch and voltage control problem are developed as follows [2]:

A. Initial Population and Parameter Selection

An initial population of control devices given in (13) is randomly generated in the parameter space using (14).

$$U_i = [V_i, T_i, nc_i] \quad i = 1, 2, \dots, np \quad (13)$$

$$u_i = u_i^{\min} + rand \cdot (u_i^{\max} - u_i^{\min}) \quad (14)$$

Where, u_i^{\min} and u_i^{\max} are the minimum and maximum values of the parameter variables, np is the population size, and $rand$ is a uniform random number generator between 0 and 1.

B. Treatment of Control Variables

Within the algorithms (PSO, MPSO, DE, and DEPSO), mixed integer nonlinear programming formulation is used. The distinction between the continuous and discrete control variables is made as follows:

- Generating units' voltage setpoints as continuous variables are assumed to operate within the range ($0.9 \leq V_{gi} \leq 1.1$).

- On-load tap changer transformers are considered to have 21 tap positions with a discrete step of 0.01 within the range ($0.9 \leq T_i \leq 1.1$).
- Number of reactors/condensers is assumed to vary between 0 and the step size (nc_i) on each bus. Each step value is also specified, e.g., for the Nigerian power system. The values of reactors are 30 MVar, 50 MVar, and 75 MVar, with step sizes of 10MVar, 16.7 MVar and 25 MVar respectively, located at 8 different busses.

C. Handling of Constraints

The reproduction operation of DE can extend the search outside the range of the parameter. A simple strategy to ensure that the parameter values lie within the allowable range after reproduction is used in this study. Any parameter that violates the limits is replaced with random values using (15).

$$u_i = \begin{cases} u_i^{\min} + \text{rand}(u_i^{\max} - u_i^{\min}), & \text{if } u_i < u_i^{\min} \text{ or } u_i > u_i^{\max} \\ u_i & \text{otherwise} \end{cases} \quad (15)$$

A penalty function approach proposed in [18] was adopted in this study to handle the voltage limits violations. The objective function is formulated according to (16):

$$f_{obj} = (P_{loss} + a) \cdot \prod_{i=1}^{nd} c_i^{b_i} \quad (16)$$

Where,

$$c_i = \begin{cases} 1 + s_i \cdot V_{Ld}, & \text{if } V_{Li} > V_{Li}^{\max} \text{ or } V_{Li} < V_{Li}^{\min} \\ 1, & \text{otherwise} \end{cases}$$

$$V_{Ld} = \begin{cases} V_{Li} - V_{Li}^{\max}, & \text{if } V_{Li} > V_{Li}^{\max} \\ V_{Li}^{\min} - V_{Li}, & \text{if } V_{Li} < V_{Li}^{\min} \end{cases}$$

$s_i \geq 1$ and $b_i \geq 1$. The constant a is used to ensure that only non-negative values are assigned to the objective function. Constant s is used for appropriate scaling of the constraint function value. The exponent b modifies the shape of the optimization surface.

D. Realization of DE Based Reactive Power Dispatch and Voltage Control

The computational procedure for the DE based approach is described as follows:

Step I: At the initialization stage, the relevant DE parameters as shown in Table I are defined. Also relevant power system data required for the computational process are actualized from the data files.

Step II: Run the base case Newton Raphson load flow [25] to determine the initial load bus voltage and active power losses respectively.

Step III: Each control device is treated as described in subsection B above. The randomly generated initial population comprises the control device variables within the parameter

space using (8). The objective function for each vector of the population is computed using (16). The vector with the minimum objective function value (the best fit) so far is determined.

Step IV: Update the generation count.

Step V: Mutation, crossover, selection, and evaluation of the objective function as described in Section III are performed. If parameter violation occurs, (8) is applied appropriately to randomly generate the parameter value. The elitist strategy is also applied to keep track of the fittest vector.

Step VI: If the generation count is less than the preset maximum number of generations, go to step IV. Otherwise the parameters of the fittest vector are returned as the desired optimum settings. With the optimal settings of the control devices, run the final load flow to obtain the final voltage profiles and the corresponding system power losses.

E. Realization of PSO Based Reactive Power Dispatch and Voltage Control

The computational procedure of the PSO based approach is described as follows:

Step I: Read the relevant PSO parameters as shown in Table I. Also relevant power system data required for the computational process are actualized from the data files.

Step II: Run the base case Newton Raphson load flow [25] to determine the initial load bus voltage and active power losses.

Step III: Each control device is treated as described in subsection B above. Then randomly generate an initial swarm of particles with random positions and velocities. Each candidate solution should be within the feasible decision variable space.

Step IV: For each individual set of control variables of the population, run the load flow to obtain the transmission losses and voltage profile. Compute the fitness values of the initial particles in the swarm using the objective function (16). Set the initial $pbest$ to the current position of each particle, and the best of the initial values among the swarm is set to $gbest$.

Step V: Increase the iteration number.

Step VI: Update the velocities and positions according to (8) and (9), respectively.

Step VII: Compute the fitness values of the new particles in the swarm using the objective function (16). Update the $pbest$ with the new positions if the particles' present fitness is better than that of the previous ones. Also update the $gbest$ with the best particle solution in the population swarm.

Step VIII: Repeat steps V to VIII until the preset convergence criterion (maximum number of generations) is achieved.

Step IX: The parameters of the $gbest$ at the end of the run are returned as the desired optimum settings. With the optimal settings of the control devices, run the final load flow to obtain the final voltage profile and the corresponding system power losses.

Table I: Optimal Parameter Settings for DE and PSO Based Approaches

DIFFERENTIAL EVOLUTION	PARTICLE SWARM
Maximum generation, $iter^{max}$: 200	Maximum generation, $iter^{max}$: 200
Population size, np : 20	Swarm size, np : 20
Scaling factor, F : 0.4	Object. Function scaling const, a : 7
Object. Function scaling const, a : 7	Constraint scaling constant, s : 1
Constraint scaling constant, s : 1	Opt. surface shape modifier, b : 1
Opt. surface shape modifier, b : 1	Cognitive constant, c_1 : 2
Crossover constant, CR : 0.6	Social constant, c_2 : 2
	Maximum inertia weight, w^{max} : 0.9
	Minimum inertia weight, w^{min} : 0.2
	Maximum velocity, v^{max} : resolution
	N: 2

F. Realization of DEPSO Based Reactive Power Dispatch and Voltage Control

The procedure for performing a DEPSO based approach combines the DE and PSO approaches described above. The relevant parameters for the DEPSO approach are provided in Table II. First, steps I through VI are performed using the PSO approach. Then step V of the DE based approach is carried out on each PSO particle to attempt to improve its solution using evolutionary techniques. In this step, a DE offspring is created for each parent PSO particle. The fitness of each parent is compared to the fitness of its offspring and the ones with the better fitness are used to complete steps VII through IX of the PSO approach.

G. Realization of MPSO Based Reactive Power Dispatch

The computational procedure for performing a MPSO based search is similar to the PSO approach described above. The relevant parameters for the MPSO approach are provided in Table II. The PSO concept is carried out using the steps previously described with the following change to step VI:

Step VI: Update the velocities and positions according to (8) and (9), respectively. If the number of iterations is greater than 75, the particles chosen for mutation as described in Section III-B are mutated and the particle's position is updated using (12).

Table II: Optimal Parameter Settings for DEPSO and MPSO Based Approaches

DEPSO	MPSO
Maximum PSO generation, $iter^{max}$: 200	Maximum generation, $iter^{max}$: 200
Maximum DE generation, $iter^{max}$: 1	Swarm size, np : 20
Swarm size, np : 20	Object. Function scaling const, a : 7
Scaling factor, F : 0.4	Constraint scaling constant, s : 1
Object. Function scaling const, a : 7	Opt. surface shape modifier, b : 1
Constraint scaling constant, s : 1	Cognitive constant, c_1 : 2
Opt. surface shape modifier, b : 1	Social constant, c_2 : 2
Crossover constant, CR : 0.6	Maximum inertia weight, w^{max} : 0.9
Object. Function scaling const, a : 7	Minimum inertia weight, w^{min} : 0.2
Constraint scaling constant, s : 1	Maximum velocity, v^{max} : resolution
Opt. surface shape modifier, b : 1	N: 2
Cognitive constant, c_1 : 2	Mutation rate, m : 0.3
Social constant, c_2 : 2	Mutation iteration start point: 75
Maximum inertia weight, w^{max} : 0.9	
Minimum inertia weight, w^{min} : 0.2	
Maximum velocity, v^{max} : resolution	
N: 2	

V. SIMULATION RESULTS AND DISCUSSION

The algorithms are implemented in MATLAB. The effectiveness of the approaches is demonstrated on the Nigerian 330 kV, 31-bus transmission grid. The simulated power system is composed of 7 generating units (4 thermal units and 3 hydro), 7 machine transformers equipped with tap changers, and compensation reactors of different discrete values located at 8 different nodes. The single line diagram of the network is depicted in Fig. 1 [5] and the network data can be obtained from [26].

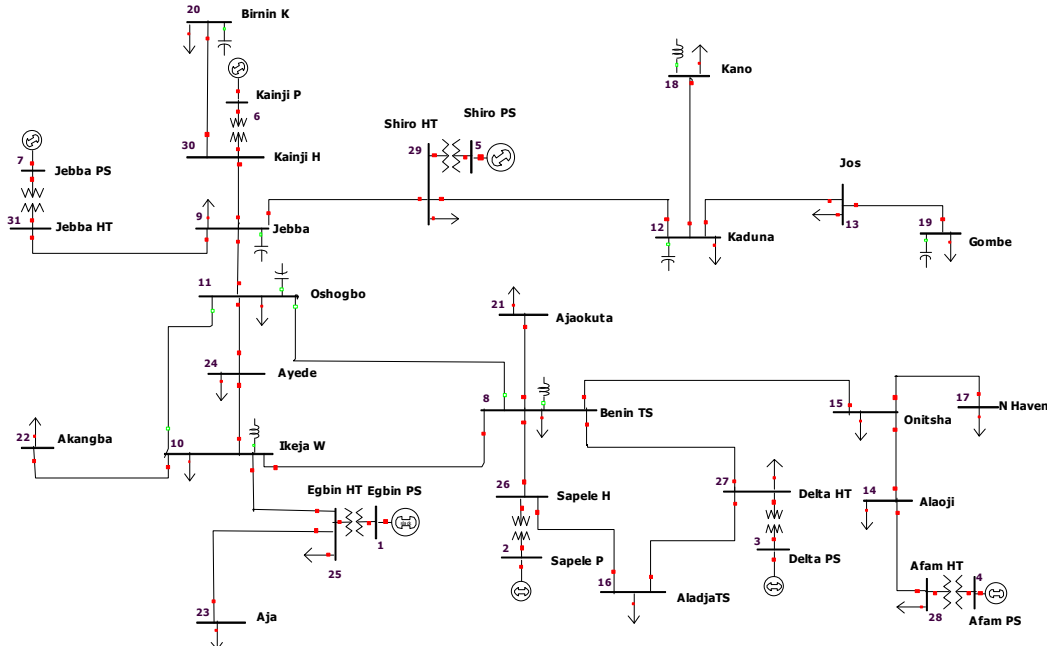


Fig 1. Single line diagram of the Nigerian 330 kV grid system.

A. Case Study 1: Wrong Tap Settings of Transformer and Inductors

The power system is preset with all 33 transmission lines in operation and wrong tap settings of the machine transformer taps. Two of the four 75 MVar reactors, at bus 8 (Benin TS) and bus 10 (Ikeja W), are wrongly switched on [2]. This setup along with load reductions at some points led to an initial power loss of 40.59 MW and 6 voltage limit violations.

All four algorithms are applied to solve this case study. The results for the four methods in terms of percent power loss reduction and the minimum number of iterations performed to achieve that percentage are provided in Table III. The results are averaged over 50 trial runs. In comparing the power loss reduction capabilities, it can be seen that the PSO based approaches are more suitable than the DE approach alone. The PSO algorithm reduced losses by 11.78% whereas the DE approach only achieved a power loss reduction of 6.69%. Combining the two algorithms in the DEPSO based approach did not improve the performance over PSO as they achieved similar power loss reduction, but the PSO results occurred in fewer iterations. Adding the mutation to PSO for the MPSO algorithm provided the best overall results with a power loss reduction of 14.17%. The energy saved per second by using the MPSO method over the PSO method is 2.39 MJ while the system topology remains in the same state.

Table III. Average number of Iterations for All Algorithms (Averaged over 50 trials)

	% Power Loss Reduction (MW)			Number of Iterations		
	Min.	Max.	Average	Min.	Max.	Average
DE	3.83	9.65	6.69	2	190	135
PSO	8.71	14.50	11.78	13	200	111
DEPSO	7.89	15.91	11.37	38	200	143
MPSO	8.71	23.06	14.17	51	200	135

The results of the voltage profile corrections averaged over 50 trials for the DE based methods and PSO based methods are presented in Figs. 2 and 3, respectively. For this case, all four algorithms brought the bus voltages within the voltage limits.

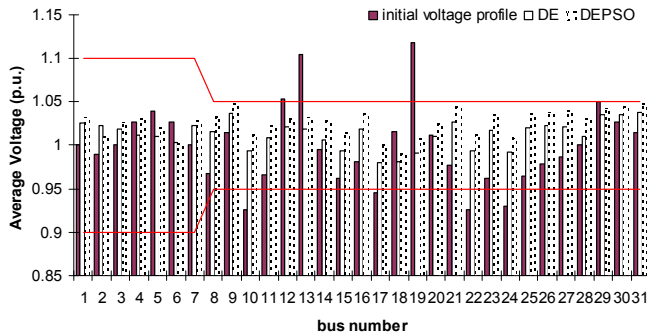


Fig. 2. Average voltage profile with DE and DEPSO for case study 1.

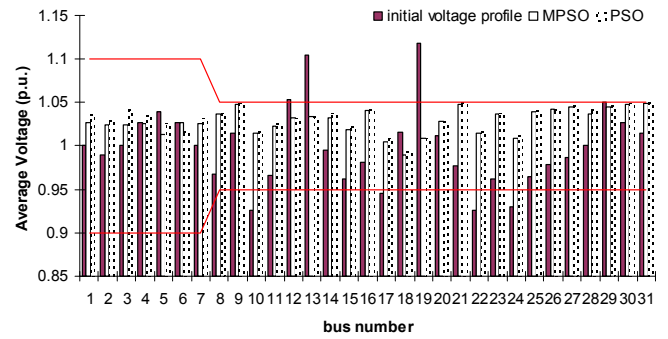


Fig. 3. Average voltage profile with MPSO and PSO for case study 1.

B. Case Study 2: Disconnection of a Transmission Line

In this case study, the system is initially set up as in case study 1 above. In addition, the transmission line between Oshogbo and Benin TS (11-8) is removed. This resulted in an initial power loss of 47.82 MW and 7 voltage limit violations.

Again, all four algorithms are applied to this case and the power loss reduction is given in Table IV. The algorithms are all successful in returning the bus voltages to acceptable levels within the stated limits.

Table IV. Average number of Iterations for All Algorithms (Averaged over 50 trials)

	% Power Loss Reduction (MW)			Number of Iterations		
	Min.	Max.	Average	Min.	Max.	Average
DE	5.94	11.35	8.47	43	199	157
PSO	6.97	15.02	12.88	29	200	136
DEPSO	9.20	16.47	12.75	26	200	134
MPSO	9.04	24.96	16.07	13	200	134

The same basic trend from the results in Case Study 1 occurred in this case study. MPSO achieved the best results with power loss reduction of 16.07 % in only 134 iterations. 3.19 MJ of energy is saved per second using the MPSO method over the PSO method while the system topology remains in this state. A plot of average power loss per iteration for all four methods is provided in Fig. 4. The results in Fig. 4 are averaged over 50 trials. The power loss convergence characteristics for PSO and DEPSO are almost identical. All three PSO based algorithms outperform DE solely based approach. In comparing the PSO and MPSO convergence results, Fig. 4 shows that by applying the mutation after the first few PSO iterations, a greater power loss percentage can be achieved.

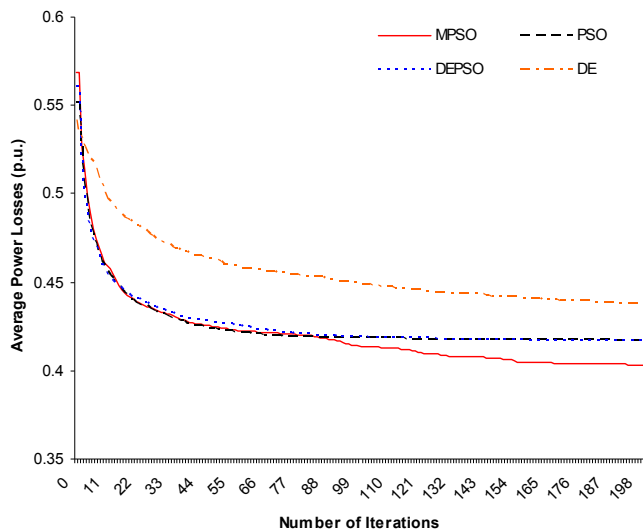


Fig. 4. Comparison of convergence characteristics for all algorithms for case study 2. (Averaged over 50 trials)

C. Discussion of Results

The case study results show that the swarm intelligence and evolutionary based approaches can successfully achieve voltage profile correction and power loss reduction within less than 200 generations. The PSO based approaches outperformed the DE approach and DEPSO did not provide much better results than performing PSO alone. Applying a slight *gbest*-oriented mutation to the PSO approach significantly improved the power loss reduction results, even only when the mutation operation is introduced in the last few iterations.

VI. CONCLUSION

This paper has presented and compared four algorithms based on swarm intelligence and evolutionary techniques for solving the optimal reactive power dispatch and voltage control problem. Case studies on the Nigerian power system illustrate the effectiveness of these algorithms in terms of the quality of the solutions found and their convergence characteristics. All four algorithms are able to successfully restore the bus voltages to prescribed limits while lowering the system transmission power losses. It is shown by averaging the results over a multitude of trial runs, that PSO indeed outperforms the DE approach on this problem when comparing power loss reduction and number of iterations required to achieve.

Since these studies are implemented in MATLAB, computation time was not in real time and further study may be done using these algorithms on a power system simulated on a real-time simulator. Future study on this problem also needs to be pursued in the area of minimizing the number of control devices to alleviate bus voltage problems. Also, pre-selection mechanisms should be applied to select the most appropriate control devices *a priori*, to reduce the computation time of the algorithms.

ACKNOWLEDGMENT

The authors acknowledge the financial support for this research study provided by the National Science Foundation (NSF) CAREER Grant ECCS # 0348221 and the US Department of Education GAANN Fellowship Program.

REFERENCES

- [1] Bakare GA, Krost G, Venayagamoorthy GK, Aliyu UO, "Differential Evolution Approach for the Reactive Power Optimization of Nigerian Grid System," *Power Engineering Society General Meeting*, 2007, pp. 1-6.
- [2] Bakare GA, Krost G, Venayagamoorthy GK, Aliyu UO, "Comparative Application of Differential Evolution and Particle Swarm Techniques to Reactive Power and Voltage Control," *Intelligent Systems Applications to Power Systems*, 2007, pp. 1 – 6.
- [3] Lee KY, Park YM, Ortiz JL, "A United Approach to Optimal Real and Reactive Power Dispatch," *IEEE Transactions on Power Apparatus and Systems*, Vol. 104, 1985, pp. 1147-1153.
- [4] Momoh JA, El-Hawary E, Adapa R, "A Review of Selected Optimal Power Flow Literature to 1993 Part I & II," *IEEE Transactions on Power Systems*, Vol. 14, No. 1, February 1999, pp. 96-111.
- [5] Bakare GA, Venayagamoorthy GK, Aliyu UO, "Reactive Power and Voltage Control of the Nigerian Grid System Using Micro-Genetic Algorithm," *Proceedings of IEEE Power Engineering Society General Meeting*, 2005.
- [6] Bakare GA, *Removal of Overload and Voltage Problems using Genetic Algorithm and Expert Systems*, Dissertation Gerhard Mercator University, Duisburg, Germany, 2001.
- [7] del Valle Y, Venayagamoorthy GK, Mohagheghi S, Hernandez JC, Harley RG, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems," *IEEE Transactions on Evolutionary Computation*, in press.
- [8] El-Sayeed MAH, "Ruled Based Approach for Real Time Reactive Power Control in Interconnected Power System," *Expert System with applications*, Vol. 14, 1998, pp. 335-360.
- [9] Krost G, Bakare GA, "A Genetic Algorithm Based Approach for Improvement in Voltage Profile and Real Power Loss Minimization," *Proceedings of the IEEE Power Tech '99 Conference*, Budapest (Hungary), August 29-September 2, 1999, Section 24, paper BP99-293-23.
- [10] Miranda V, Srinivasan D, Proenca L, "Evolutionary Computation in Power Systems," *Electrical Power and Energy Systems*, Vol. 20, No. 2, 1998, pp. 89-98.
- [11] Yoshida H, Kawata K, Fukuyama Y, Nakanishi Y, "A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Stability Assessment," *IEEE Transactions on Power Systems*, Vol. 15, No. 4, November 2000, pp. 1232-1241.
- [12] Higashi N, Iba H, "Particle Swarm Optimization with Gaussian Mutation," *Proceedings, IEEE Swarm Intelligence Symposium*, 2003, pp. 77-79.
- [13] Miranda V, Fonseca N, "EPSO – Best-of-two-worlds Meta-heuristic applied to Power System Problems," *Proceedings, IEEE Congress on Evolutionary Computation*, 2002, pp. 1080-1085.
- [14] Stacey A, Jancic M, Grundy I, "Particle Swarm Optimization with Mutation," *Proceedings, IEEE Congress on Evolutionary Computation*, 2003, pp. 1425-1430.
- [15] Wei C, Zheng Y, Pi W, "Swarm Directions Embedded in Fast Evolutionary Programming," *Proceedings, IEEE Congress on Evolutionary Computation*, 2002, pp. 1278-1283.
- [16] Saadat H, *Power System Analysis*, Tata McGraw-Hill Publishing Company Ltd. New Delhi, India.
- [17] Storn R, Price K, "Differential Evolution – a Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces," Technical Report TR-95-012, *International Computer Science Institute*, Berkeley, CA, March 1995.
- [18] Lampinen J, Zelinka I, "Mixed Integer-Discrete-Continuous Optimization by Differential Evolution," *Proceedings of the 5th International Conference on Soft Computing*, 1999, pp. 71-76.

- [19] Kennedy J, Eberhart R, "Particle Swarm Optimization," *Proceedings, IEEE International Conference on Neural Networks*, 1995, Vol. 4, pp. 1942 – 1948.
- [20] Kennedy J, Eberhart R, Shi Y, *Swarm Intelligence*, Morgan Kaufman Publishers, 2001.
- [21] Zhang WJ, Xie F, "DEPSO: Hybrid Particle Swarm with Differential Evolution Operator," *IEEE International Conference on Systems, Man and Cybernetics*, Washington DC, USA, 2003, pp. 3816-3821.
- [22] Esmiin AAA, Lambert-Torres G, Zambroni de souza AC, "A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization," *IEEE Transactions on Power Systems*, Vol. 20, No. 2, 2005, pp. 859-866.
- [23] Lu ZS, Hou ZR, "Particle Swarm Optimization with Adaptive Mutation," *Acta Electronica Sinica*, Vol. 32, No. 3, 2004, pp. 416-420.
- [24] Yare Y, Venayagamoorthy GK, "Optimal Scheduling of Generator Maintenance using Modified Discrete Particle Swarm Optimization," *Bulk Power System Dynamics and Control Symposium-VII*, 2007, pp. 1 – 8.
- [25] Zimmermann RD, Murillo-Sanchez CE, Gan D, "MATLAB Power System Simulation Package," Version 3.0.0, February 2005.
- [26] Wudil TS, Loss Formula Computation for Nigerian Grid System, M. Eng. Degree Thesis, A. T. B. University, Bauchi, Nigeria, 2004.