# Comparative Application of Differential Evolution and Particle Swarm Techniques to Reactive Power and Voltage Control

G. A. Bakare, G. Krost, *Member IEEE*, G. K. Venayagamoorthy, *Senior Member IEEE*, and U. O. Aliyu, *Member IEEE*

***Abstract*-- This paper presents the comparative application of two metaheuristic approaches: Differential Evolution (DE) and Particle Swarm Optimization (PSO) to the solution of the reactive power and voltage control problem. Efficient distribution of reactive power in an electric network leads to minimization of the system losses and improvement of the system voltage profile. It can be achieved by varying the excitation of generators or the on-load tap changer positions of transformers as well as by switching of discrete portions of inductors or capacitors etc. This constitutes a typical mixed integer non-linear optimization problem for the solution of which metaheuristic techniques have proven well suited in principle. The feasibility, effectiveness and generic nature of both DE and PSO approaches investigated are exemplarily demonstrated on the Nigerian grid system and the New England power system. Comparisons were made between the two approaches in terms of the solution quality and convergence characteristics. The simulation results revealed that both approaches were able to remove the voltage limit violations, but PSO procured in some instances slightly higher power loss reduction as compared with DE; on the other hand DE required a lower number of function evaluations as compared with PSO. Consideration of computational effort is relevant for potential real time on line application.**

***Index Terms*--Reactive power / voltage control, Differential evolution, Particle swarm optimization, Metaheuristic.**

## I. INTRODUCTION

Due to the steady increase in the complexity of power systems and the continuous high loading of network components, abnormal operating conditions such as under voltage may occur more frequently. Hence, the need for appropriate reactive power and voltage control of the power system is evident. The reactive power dispatch has two-fold objectives thus: to improve the system voltage profile and to minimize system losses at all times. The reactive power flow can be controlled by suitably adjusting the following facilities:

- 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 line.

The foregoing control devices have their lower and upper permissible limits and are distributed system-wide. It is therefore evident that the reactive power and voltage control problem for a real large power system is very complex encompassing different control devices, some of which are continuously adjustable whilst others are of discrete steps that are numerous, asymmetrical and geographically located dispersed. The existence of multiple optimum solutions is inevitable most especially when there are many reactive power control devices to be manipulated in order to secure desired target system voltages in a typically large power system. Thus, there is a need to develop intelligent technology to achieve the global optimum solution of the reactive power dispatch problem.

Several numerical optimization techniques have been proposed within the framework of optimal power flow to assist the operator in reaching the optimal decision. Among these techniques, Nonlinear Programming (NLP), successive linear programming, mixed integer programming, Newton and quadratic techniques have been proposed for solving the Var control problem [1]. The drawbacks of these techniques have been extensively discussed in [2]. In an attempt to circumvent the deficiencies of the conventional methods, several search techniques have been proposed; they are Expert System (ES), Genetic Algorithm (GA), Tabu Search (TS), Simulated Annealing (SA), Evolution Strategy (ES), Particle Swarm Optimization (PSO), etc. [2 - 8].

In this paper, two metaheuristic techniques: Differential Evolution (DE) and Particle Swarm Optimization (PSO) are explored as optimization tools for controlling the reactive power for improvement of the voltage profiles and reduction of system losses. Generators, on-load tap changer positions of transformers and shunt inductors were considered as reactive power control devices in this study.

*Differential Evolution* is an improved version of GA for faster optimization [9]. The main advantages are simple structure, ease of use, robustness and effectiveness. As a robust and powerful adaptive tool for solving search and optimization problems they have been proposed for various power system problems such as generation expansion [11], capacitor placement [12], etc..

This work was supported by German Academic Exchange Service (DAAD), Germany under the re-invitation program (http://www.daad.de).

G. A. Bakare and U. O. Aliyu are with Electrical and Electronics Engineering Programme, Abubakar Tafawa Balewa University, Bauchi, Nigeria (e-mail: bakare_03@yahoo.com and uoaliyu@yahoo.com).

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

G. K. Venayagamoorthy is with Real-Time Power and Intelligent Systems Laboratory, University of Missouri-Rolla, MO 65409 USA (e-mail: gkumar@ieee.org).

*Particle Swarm Optimization* is an evolutionary computation technique which has been found to be robust in solving continuous nonlinear optimization problems [5,8]. The PSO algorithm is simple in concept, easy to implement and computationally efficient.

The DE and PSO tools for reactive power and voltage control of power system have been developed using MATLAB Version 7.1 R14 and were demonstrated on two networks: the Nigerian transmission grid and the New-England system modeled on the power world simulator in detail. This provides a platform to preset a multitude of scenarios under operational realism.nomenclature list, if needed, should precede the Introduction.

## II. Problem Formulation

The mathematical model for the optimal reactive power / voltage control problem is formulated as follows:

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

subject to

$$G(X,U)=0 \tag{2}$$

$$H(X,U)\geq 0 \tag{3}$$

$$X^{\min} \leq X \leq X^{\max}$$

$$U^{\min} \leq U \leq U^{\max}$$

Where: $P_j$ are the real power losses in line $j$, $nl$ is the number of transmission lines.

$$\begin{aligned} X^T &= \left[V_{L1}, V_{L2} \ldots V_{L_{nd}}, Q_{g1}, Q_{g2} \ldots Q_{g_{ng}}\right] \\ U^T &= \left[V_{g_1}, V_{g_2} \ldots V_{g_{ng}}, T_1, T_2 \ldots T_{nt}, Q_{C_1}, Q_{C_2} \ldots Q_{C_{nc}}\right] \end{aligned} \tag{4}$$

$X$ is the vector of dependent variables, comprising load bus voltages $V_L$ and generator reactive power outputs $Q_g$. $U$ is the vector of control variables, comprising generator voltages $V_g$, transformer tap settings $T$, and shunt Var compensation $Q_C$. $G(X,U) = 0$ and $H(X,U) \geq 0$ are the typical load flow equations [13]. They are solved using the Newton Raphson load flow.

## III. Metaheuristics Concept

### A. Differential Evolution

Differential evolution was introduced by Storn and Price in 1995 as heuristic optimization method which can be used to minimize nonlinear and non-differentiable continuous space functions with real-valued parameters [9]; it uses floating point numbers to encode the parameter variables in contrast with conventional GA that uses binary coding. It has been extended to handle mixed integer discrete continuous optimization problems, too [10]. Design principles in DE's are [9]:

- Simple structure, ease of use and robustness.
- Operating on floating point format with high precision.
- Effective for integer, discrete and mixed parameter optimization.
- Handling non-differentiable, noisy and/or time dependent objective functions.
- Effective for nonlinear constraint optimization problems with penalty functions, etc..

Like the other evolutionary algorithm family, DE also relies on an initial random population generation, which is then improved using selection, mutation, and crossover repeated through generations until the convergence criterion is met.

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

$$U_3' = U_3 + F \cdot (U_1 - U_2) \tag{5}$$

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 the risk of being stuck at a local optimum. The crossover is a complementary process for DE. It aims at reinforcing the prior successes 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 $U_t$ according to (6):

$$U_t = \begin{cases} U_3' & \text{if rand} < CR \\ U_i & \text{otherwise} \end{cases} \tag{6}$$

Where: $CR$ is a crossover constant and its value is typically $(0 \leq CR \leq 1.0)$. 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:

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

### B. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Kennedy and Eberhart [5,8]. The method is derived from simulation of a simplified social model of swarms such as fish schooling and bird flocking, is based on a simple concept, has been found to be robust for solving problems featuring non-linearity and non-differentiability, multiple optima and high dimensionality through adaptation, and provides high quality solutions with stable convergence.

The individuals (particles) persist over time, influencing one another's search of the problem space, as compared with genetic algorithms where the weakest chromosomes are immediately discarded. Instead of using evolutionary operators to manipulate the individuals as in other evolutionary computation algorithms, each individual in the swarm flies in the search space with a velocity which is dynamically adjustable according to its own flying experience (velocity, inertia, gravity) and its companion flying experience. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called *pbest*. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population. This is called *gbest*. The basic concept of PSO technique lies in accelerating each particle towards its *pbest* and *gbest* locations at each time step. The modified velocity of each particle can be computed using the current velocity and the distance from *pbest* and *gbest* according to (8). The positions are modified using (9).

$$v_{id}^{k+1} = w^k \cdot v_{id}^k + c_1 \cdot rand_1 \cdot (pbest_{id} - x_{id}^k) + c_2 \cdot rand_2 \cdot (gbest_d - x_{id}^k) \qquad (8)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \qquad (9)$$

Where:

$rand_1, rand_2$: uniformly random numbers between 0 and 1.
$v^k_{id}$: current velocity of individual $i$ in dimension $d$ at iteration $k$.
$v^{k+1}_{id}$: velocity of individual $i$ in dimension $d$ at iteration $k+1$.
$v_{id}^{\min} \le v_{id}^k \le v_{id}^{\max}$ : maximum and minimum velocity.
$x^k_{id}$: current position of individual $i$ in dimension $d$ at iteration $k$.
$x^{k+1}_{id}$: 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 particles 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).

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

$iter^{max}$ : 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 the fitness of search depend on $v^{max}$. If $v^{max}$ is too high, then particles will move in larger steps and so the solution reached may not be optimal. If the $v^{max}$ is too low, then 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).

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

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

## IV. Realization of the Metaheuristic Tools

Both the differential evolution and particle swarm based reactive power and voltage control tools were developed as follows:

### *A. Initial Population and Parameters Selection*

For both methods, an initial population comprising control devices

$U_i=[V_i, T_i, nc_i]$ ; $i=1,2,....np$

is randomly generated within the parameter space using (12).

$$u_i = u_i^{\min} + rand \cdot (u_i^{\max} - u_i^{\min}) \qquad (12)$$

Where: $u_i^{min}$ and $u^{max}$ are respectively the minimum and the maximum values of the parameter variables, $np$ is the population size and $rand$ is a uniform random number generator in [0, 1].

### *B. Treatment of Control Variables*

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

- Generating units' voltage set-points as continuous variables are assumed to operate within the range ($0.9 \le V_{gi} \le 1.1$).
- On-load tap changer transformers are considered to have 21 tap positions with a discrete step of 0.01 within the range ($0.9 \le T_i \le 1.1$).
- The 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 grid system, the values of reactors are 30 MVar, 50 MVar and 75 MVar with step sizes ranging between 1 and 4 located at 8 different buses.

### *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 was adopted in this study. Any parameter that violates the limits is replaced with random values using (13).

$$u_i = \begin{cases} u_i^{\min} + rand(u_i^{\max} - u_i^{\min}) & \text{if } u_i < u_i^{\min} \text{ or } u_i > u_i^{\max} \\ u_i \text{ otherwise} \end{cases} \qquad (13)$$

A penalty function approach proposed in [10] was adopted in this study to handle the voltage limits violations. The objective function is formulated according to (14):

$$f_{obj} = (P_{loss} + a) \cdot \prod_{i=1}^{nd} c_i^{b_i} \qquad (14)$$

Where:

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

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

$s_i \geq 1$ 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

The computational procedure of the developed tool 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 [14] to determine the initial load bus voltage and active power losses respectively.

*Step III:* Each control device is treated as described in sub-section B above. The randomly generated initial population comprises the control device variables within the parameter space using (12). The objective function for each vector of the population is computed using (14). The vector with the minimum objective function value (the best fit) so far is determined.

*Step IV*: Update of 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, (12) is applied appropriately to generate randomly the parameter value. The elitist strategy is also applied: keeping 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

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 [14] to determine the initial load bus voltage and active power losses respectively.

*Step III:* Each control device is treated as described in sub-section 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 (14). Set the initial *pbest* to current position of each particle, and the initial best evaluated values among the swarm is set to *gbest*.

*Step V*: Increase the generation 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 (14). 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 in the population swarm.

*Step VIII*: Repeat steps V to VII 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

| DIFFERENTIAL EVOLUTION | PARTICLE SWARM |
|---|---|
| Maximum generation, $iter^{max}$: *200* | Maximum generation, $iter^{max}$: *200* |
| Population size, *np: 15* | Swarm size, *np: 50* |
| 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 for Nigerian grid & 5 for New-England system |

## V. Simulation Results and Discussion

The above described procedures for both DE and PSO were implemented using MATLAB V 7.1 R14 for Windows. The feasibility, effectiveness and generic nature of the approaches were demonstrated on two power systems: Nigerian 330 kV, 31-bus transmission grid and New England 39-bus, 10 machines system.

### A. Example1: Nigerian 330 kV Grid System

The replicated power system comprises: 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 [2] and the network data can be obtained from [15]. For comparison purpose, two samples of the multitudes of studies conducted on this power system using both DE and PSO are presented here. The results are compared in terms of convergence characteristics and solution quality.

### A.I. Case Study 1: Tap Settings of Transformer and Inductor

With all the 33 transmission lines operated, a scenario was preset on the power world simulator by heuristic based wrong settings of the machine transformer taps. Furthermore, two 75 MVar reactors at different buses were wrongly switched on. There were also load reductions at some load points. These actions altogether led to voltage limit violations at 10 nodes.

The developed DE and PSO tools were applied to solve this problem with the parameter settings as shown in Table I. The results of the voltage profile corrections for both methods are comparatively shown in Fig. 1. It can be seen that both approaches were able to bring the voltage at all buses within the limits. The convergence characteristics of the two methods are comparatively depicted in Fig. 2.

It can be seen that a power loss reduction of 13.55% (from 40.07 MW to 34.64 MW) was accomplished in 162 generations using the DE approach while the PSO achieved 16.62% (from 40.07 MW to 33.41 MW) in 193 generations.

The corresponding total number of function evaluations to obtain the minimum power losses are 2,230 and 9,650 for DE and PSO respectively. It can be seen that DE requires remarkably less function evaluations since the population size was set at 15 while it was 50 for PSO.

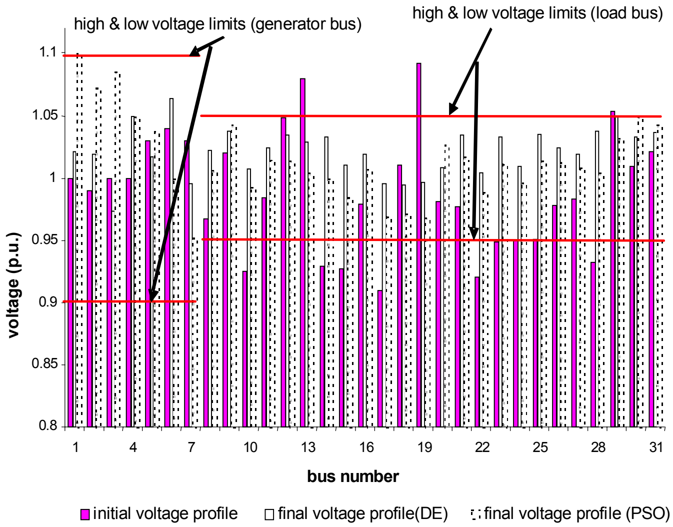


Fig. 1: Voltage profile correction for DE and PSO

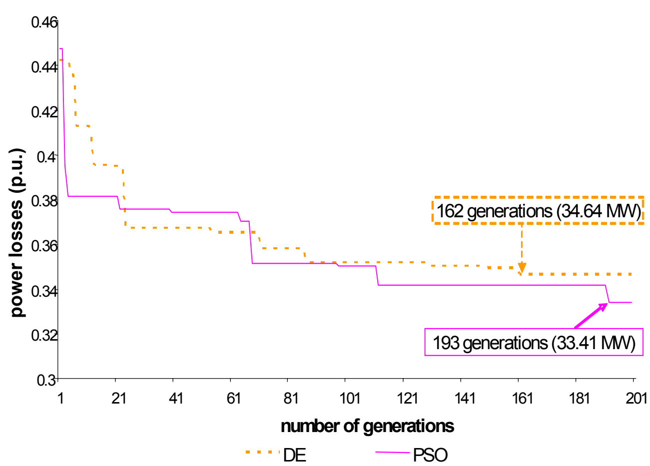


Fig. 2: Comparison of convergence characteristics for DE and PSO

*A.II. Case Study 2: Disconnection of a transmission line*

Here, the system was initially operating as in scenario 1. Interrupting a transmission line broke up a mesh and resulted in voltage limits violations at 12 nodes.

The developed reactive power dispatch tools were used to solve this problem. The voltage profile corrections for both methods are comparatively shown in Fig. 3. Both approaches succeeded in solving the voltage problem connected with 15.34% power loss reduction (from 42.05 MW to 35.60 MW) for DE while the PSO achieved 15.15% power loss reduction (from 42.05 MW to 35.68 MW). These values were obtained in 185 and 174 generations for both DE and PSO respectively, Fig. 4. The required numbers of function evaluations were 2,775 for DE and 8,700 for PSO.

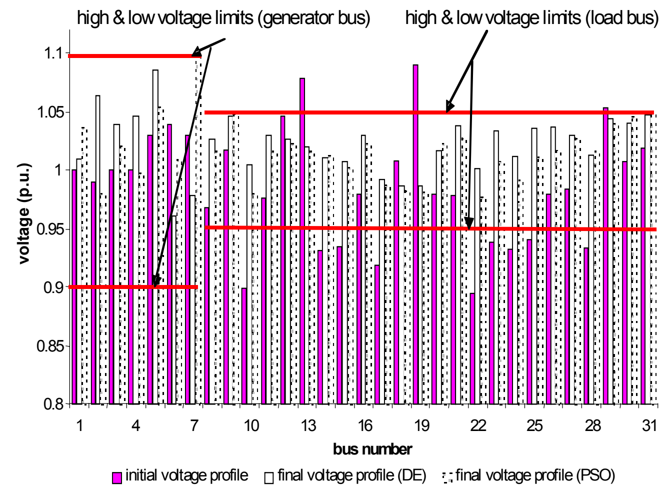


Fig. 3: Voltage profile correction for DE and PSO (Case study 2)

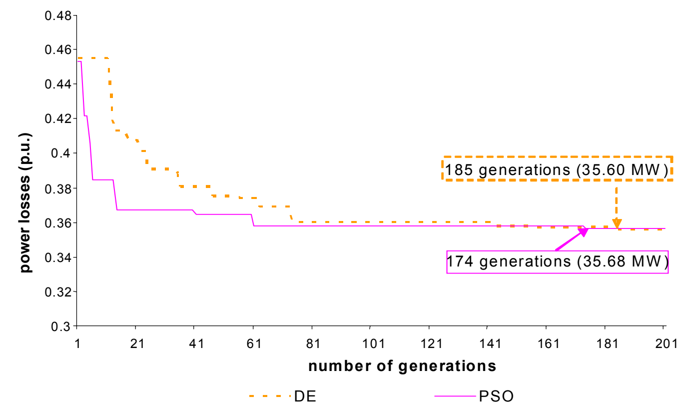


Fig. 4: Convergence characteristics for DE and PSO (Case study 2)

*B. Example 2: New England System*

This is a 39-bus system and has also been simulatively replicated. It comprises: 10 generating units, 12 transformers equipped with tap changers for voltage control. A scenario was preset on the simulator that led to high voltage limit violations on three buses. The results of the application of the both DE and PSO based tools for voltage corrections and loss reductions are shown in Figs. 5 & 6 respectively.

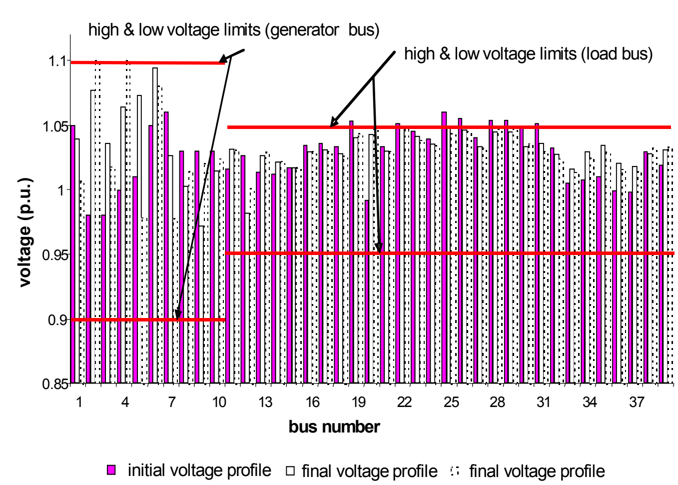


Fig. 5: Voltage profile correction for DE and PSO (New-England system)

It can be seen from these figures that both approaches succeeded in solving the voltage problem connected with 1.60% power loss reduction (from 42.01 MW to 41.34 MW) for DE while the PSO achieved 4.14% power loss reduction (from 42.01 MW to 40.07 MW). These values were obtained in 188 and 109 generations and required 2,820 and 5,450 function evaluations for both DE and PSO respectively.

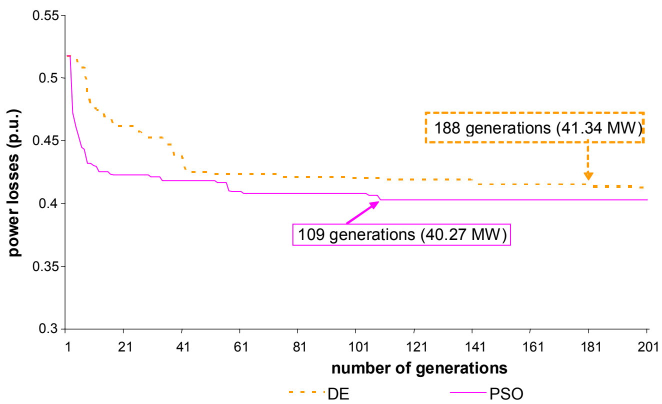


Fig. 6: Convergence characteristics for DE and PSO (New-England system)

## *C. Discussion of Results*

Results of the case studies on the two power systems considered here have shown that both approaches succeeded in achieving the goals of voltage profile correction and power loss reduction within less than 200 generations. In the implementation phase few preparative calculations sufficed to find appropriate parameter settings for both approaches; DE generally requires less settings than PSO (see Table I). PSO procured little better power loss reduction in some cases as compared with DE but connected with considerably higher number of function evaluations (load flow calculations) which strongly influences the overall computation effort and time. Regarding the performance it must be respected that both approaches compared here are based on stochastic search; thus, definite statements concerning the computational efficiency would require systematic investigation based on a multitude of test runs. Furthermore, the influence of swarm size and population size respectively could also be studied in order to achieve optimal computational efficiency.

## VI. Summary and Conclusions

In this paper, both Differential Evolution (DE) and Particle Swarm (PSO) based reactive power and voltage control have been comparatively investigated on two networks. The simulation results revealed that both approaches were able to remove the voltage limit violations, but PSO procured in some instances slightly higher power loss reduction as compared with DE. On the other hand, in the observed test cases DE required a considerably lower number of function evaluations as compared with PSO; if this observation could be substantiated by further investigation, the DE approach seems more viable for potential real time application in a control centre where the computation time is most relevant.

From the practical point of view, it is pertinent to curtail the number of control devices employed to alleviate bus voltage problems. It is also feasible to integrate a pre-selection mechanism or the sensitivity matrix into the algorithms to select the most appropriate control devices *a priori*, thus bringing an added advantage to the computational time of the algorithms. This will be pursued in subsequent research thrust.

## References


[1] J. A. Momoh, M. E. El-Hawary and R. Adapa, "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.

[2] G. A. Bakare, G. K. Venayagamoorthy and U. O. Aliyu "Reactive Power and Voltage Control of the Nigerian Grid System Using Micro-Genetic Algorithm", Proceedings of IEEE Power Engineering Society General Meeting, 2005.

[3] M. A. H. El-Sayeed, "Ruled Based Approach for Real Time Reactive Power Control in Interconnected Power System", *Expert System with applications,*, 14, 1998, pp. 335-360.

[4] V. Miranda, D. Srinivasan and L. Proenca, "Evolutionary Computation in Power Systems", *Electrical Power and Energy Systems,* Vol. 20, No. 2, 1998, pp. 89-98.

[5] H. Yoshida K. Kawata Y. Fukuyama and Y. Nakanishi, "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.

[6] G. Krost and G. A. Bakare, "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.

[7] G. Bakare, *Removal of Overload and Voltage Problems using Genetic Algorithm and Expert Systems*, Dissertation Gerhard Mercator University, Duisburg, Germany, 2001.

[8] Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, R.G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", *IEEE Transactions on Evolutionary Computation*, in press.

[9] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", *Technical Report TR-95-012,* March 1995.

[10] J. Lampinen and I. Zelinka, "Mixed Integer-Discrete-Continuous Optimization by Differential Evolution" *http://citeseer.ist.psu.edu/319741.html*.

[11] S. Kannan, S. M. R. Slochanal and N. P. Pandhy, "Application and Comparison of Metaheuristic Techniques to Generation Expansion Planning +Problem", *IEEE Trans. on Power Systems*, Vol. 20, No. 1, February 2003, pp. 466-475.

[12] J. P. Chiou, C. F. Chang and C. T. Su, "Ant Direct Hybrid Differential Evolution for Solving Large Capacitor Placement Problems", *IEEE Trans. on Power Systems*, Vol. 19, No. 4, November 2004, pp. 1794-1800.

[13] H. Saadat, *Power System Analysis,* Tata McGraw-Hill Publishing Company Ltd. New Delhi, India.

[14] R. D. Zimmermann, C. E. Murillo-Sanchez and D. Gan, "MATLAB Power System Simulation Package", Version 3.0.0, February 2005.

[15] T. S. Wudil, Loss Formula Computation for Nigerian Grid System, M. Eng. Degree Thesis, A. T. B. University, Bauchi, Nigeria, 2004.