J. Automation & Systems Engineering 9-4 (2015): 255-265

Regular paper

Optimal reactive power dispatch based on particle swarm optimization

Vivek Kumar Jain , Upendra Prasad

Electrical Engineering Department, B.I.T., Sindri, Dhanbad, Jharkhand, India

Journal of Automation & Systems Engineering

Abstract- In this paper, optimal reactive power dispatch based on particle swarm optimization approach. The objectives are power losses in transmission lines and voltage deviation of the system. The algorithm changed the stochastic initialization and adopted a principle of particle searching by itself. More than a few particles in feasible solutions were used to lead swarms motion and update the performance of the proposed approach is demonstrated with the IEEE-26 bus test system. It is observed that the reactive power has decreased while active power has increased and the simulation results show that the particle swarm optimization, which had been adjusted parameters, is better convergent time than other optimization methods.

Keywords: particle swarm algorithm, reactive power dispatch, power loss minimization and voltage deviation.

1. INTRODUCTION

The main purpose of optimal reactive power dispatch problem is to find the optimization settings of given power system that minimize an the objectives are power losses in transmission lines and voltage deviation of the system while at the some time satisfying various equality and inequality constraints [1]. The equality constrants are the power flow equations, while inequality constraints are the limits on the control variables and operating limits of the power system dependant variables. Up to now, number of techniques ranging from classical techniques like Linear programming (LP), non-linear programming and gradient based techniques have been proposed in the journalism for solving RPD problems. However, due to the approximations introduced by linearized models, the LP results may not represent the optimal solution for inherently non-linear objective functions such as the one used in the reactive power dispatch problem. However, these techniques have severe limitations in handling nonlinear, discontinuous functions and constraints, and function having multiple local minima. Regrettably, the original reactive power problem does have these Properties [2].

Simulations and the results obtained using the particle swarm optimization (PSO) algorithm based approaches were found to be better than the results obtained using the conventional method. Although, these works have solved the reactive power dispatch (RPD) problem successfully [3], none of them has considered the line flow and voltage stability constraints, which are important for any practical implementation of reactive power dispatch. PSO algorithm is an optimization tool based on population, which starts from random solution and finds an optimal value via iteration. The rule of the PSO algorithm is evaluating whether the solution is good or bad through a fitness function, and the PSO algorithm constantly adjusts position and speed and finds the global best by following the currently searched optimal value [6]. The PSO algorithm is simple and effective and has high precision, quick convergence and profound intelligent background, shows advantages in solving practical problems, and is suitable for scientific research and engineering application [7].

The proposed approach has been examined and tested on the standard IEEE 26-bus test system. The potential and effectiveness of the proposed approach are demonstrated and good quality solution. Finally, the simulation results proof that particle swarm optimization approach success for solving the reactive power dispatch (RPD) problem.

2. PROBLEM FORMULATION

The reactive power dispatch problem is to minimize the objective function for optimal settings of control variables, while satisfying various equality and inequality constraints, which can be described as follows:

2.1 Minimization of system power losses

The minimization of system real power losses (MW) can be calculated as follows:

$$\text{Min } f_1 = P_{Loss} = \sum_{K=1}^{nl} g_k\left[V_i^2 + V_j^2 - 2V_iV_j\cos(\delta_i - \delta_j)\right] \quad (1)$$

Where nl is the number of transmission lines; g_k is the conductance of the k^{th} line; V_i and V_j are the voltage magnitude at the end buses I and j of the k^{th} line, respectively, and δ_i and δ_j are the voltage phase at the end buses i and j.

2.2 Voltage deviation

Bus voltage is one of the most important security and service quality indices. Improving voltage profile can be obtained by minimizing the load bus voltage deviation from 1.0 per unit.

$$\text{Min } f2 = VD = \sum_{i=1}^{NL}\left|V_i - V_i^{ref}\right| \quad (2)$$

2.3 Minimization of fuel cost

$$f_2 = min.\sum_{i=1}^{NG}(a_iP_{Gi}^2 + b_iP_{Gi} + c_i) \quad (3)$$

Where a_i, b_i, and c_i are the cost coefficient of the i^{th} generator.

2.4 System constraints

The real power loss given by (1) is a non-linear function of bus voltages and phase angles which are a function of control variables. The minimization problem is subjected to the following equality and inequality Constraints:

2.4.1 Real Power Constraints:

$$P_{Gi} - P_{Di} - V_i\sum_{j=1}^{N_B} V_j\left(G_{ij}Cos(\delta_i - \delta_j) + B_{ij}Sin(\delta_i - \delta_j)\right) = 0, \quad i = 1,2 \ldots\ldots N_B \quad (4)$$

2.4.2 Reactive Power Constraints:

$$Q_{Gi} - Q_{Di} - V_i\sum_{j=1}^{N_B} V_j\left(G_{ij}Sin(\delta_i - \delta_j) + B_{ij}Cos(\delta_i - \delta_j)\right) = 0, \quad i = 1,2 \ldots . N_B \quad (5)$$

2.4.2 Bus Voltage magnitude contraints:

$$V_i^{min} \le V_i \le V_i^{max} \quad ; \quad i \in N_B \quad (6)$$

2.4.3 Transformer Tap position contraints:

$$t_k^{min} \leq t_k \leq t_k^{max} \quad ; \quad i \in N_T \tag{7}$$

2.4.4 Generator bus reactive power constraint:

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max} \quad ; \quad i \in N_g \tag{8}$$

2.4.5 Reactive power source capacity constraints:

$$Q_{ci}^{min} \leq Q_{ci} \leq Q_{ci}^{max} \; ; \; i \in N_c \tag{9}$$

2.4.6 Transmission line flow constraints:

$$|s_1| \leq |s_1^{max}| \tag{10}$$

$$S_l \leq S_l^{max} \quad ; \quad l \in N_l \tag{11}$$

The symbol used is follows:

t_k= Tap setting of transformer at branch k

Q_{Ci}= Reactive power generated by i^{th} capacitor bank

Q_{gi} = Reactive power generation at bus i

S_l = Apparent power flow through the i^{th} branch

N_B = Total number of buses

g_k = Conductance of buses

N_T= Number of tap-setting transformer branches

N_c = Number of capacitor banks

N_g = Number of generator buses

$$X_i^{lim} = X_i^{max} \; if \; X_i > X_i^{max} \tag{12}$$

$$X_i^{lim} = X_i^{min} \; if \; X_i < X_i^{min} \tag{13}$$

3. PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization algorithm, which is modified for optimizing complicated numerical functions and based on metaphor of human social interaction, is capable of mimicking the ability of human societies to process knowledge [8]. It has roots in two main component methodologies: artificial life (such as bird flocking, fish schooling and swarming); and, evolutionary computation. Its key concept is that potential solutions are flown through hyperspace and are accelerated towards better or more optimum solutions. Its paradigm can be implemented in simple form of computer codes and is computationally inexpensive in terms of both memory requirements and speed. It lies anywhere in between evolutionary programming and the genetic algorithms. As in evolutionary computation paradigms, the concept of fitness is employed and candidate solutions to the problem are termed particles or sometimes individuals, each of which adjusts its flying based on the flying experiences of both itself and its companion. It keeps track of its coordinates in hyperspace which are associated with its earlier best fitness solution, and also of its counterpart corresponding to the overall best value acquired thus far by any other particle in the population. Vectors are taken as presentation of particles since most optimization problems are convenient for such variable presentations.

In fact, the elemental principles of swarm intelligence are flexibility, diverse response, proximity, quality, and stability. It is adaptive corresponding to the change of the best group value. The allocation of responses between the individual and group values ensures a diversity of response. The higher dimensional space calculations of the PSO concept are performed over a series of time steps. The population is responding to the excellence factors of the previous best individual values and the previous best group values. The principle of stability is adhered to since the population changes its state if and only if the best group value changes [9]. As it is reported in [10], this optimization technique can be used to solve many of the same kinds of problems as GA, and does not suffer from some of GAs difficulties. It has also been found to be robust in solving problem featuring non-linear, non-differentiability and high-dimensionality. PSO is the search method to improve the speed of convergence and find the global optimum value of fitness function.
The PSO starts with a population of random solutions ‘‘particles’’ in a D-dimension space. The i^{th} particle is represented by $X_i = (x_{i1}, x_{i2}. \ . \ . \ x_{iD})$. Each particle keeps track of its coordinates in hyperspace, which are associated with the fittest solution it has achieved so far. The value of the fitness for particle i (p_{best}) am also stored as $Pi = (p_{i1}, p_{i2...} \ p_{iD})$. The global version of the PSO keeps track of the overall best value (gbest), and its location, obtained thus far by any particle in the population. PSO consists of changing the velocity of each particle toward its pbest and gbest, at each step according to Eq. (11). The velocity of particle i is represented as $V_i = (v_{i1}, v_{i2...}$ viD). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and gbest. The position of the i^{th} particle is then updated according to Eq. (10). One modification is to introduce a local-oriented paradigm (l_{best}) with different neighborhoods. It is concluded that gbest version performs best in terms of median number of iterations to converge [11]. However, Pbest version with neighborhoods of two is most resistant to local minima. PSO algorithm is further improved via using a time decreasing inertia weight, which leads to a reduction in the number of iterations. Fig.2 shows the flowchart of the proposed PSO algorithm [10]. This new approach features many advantages; it is simple, fast and easy to be coded.

Also, its memory storage requirement is minimal. Moreover, this approach is advantageous over evolutionary and genetic algorithms in many ways. First, PSO has memory. That is, every particle remembers its best solution (local best) as well as the group best solution (global best). Another advantage of PSO is that the initial population of the PSO is maintained, and so there is no need for applying operators to the population, a process that is time and memory-storage-consuming [8, 9].

3.1 Particle swarm model for continuous variables

In Particle Swarm Optimization, the particles are “flown” through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) that it has achieved so far. This implies that each particle has a memory, which allows it to remember the best position on the feasible search space that it has ever visited. This value is commonly called previous best (p-best). Another best value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the neighborhood of the particle. This location is commonly called global best (g-best). The basic concept behind the Particle Swarm Optimization technique consists of changing the velocity (or accelerating) of each particle toward its p-best and the g-best positions at each time step. This means that each particle tries to modify its current position and velocity according to the distance between its current position and p-best, and the distance between its current position and g-best. In its canonical form, Particle

Swarm Optimization is modeled as follows:

$$V_i^{k+1} = V_i^k + C_1 rand_1(.) \times (p_{best_i} - s_i^k) + C_2 rand_2(.) \times (g_{best_i} - s_i^k) \quad (14)$$

Where,

v_i^k : velocity of agent i at iteration k, c_j, : weighting factor, rand : uniformly distributed random number between 0 and 1, s_i^k : current position of agent i at iteration k, p_{besti} : p_{best} of agent i, g_{best}: g_{best} of the group.

The current position (searching point in the solution space) can be modified by the following equation

$$s_i^{k+1} = s_i^k + V_i^{k+1} \quad (15)$$

Expressions in equations (14) and (15), describe the velocity and position update, respectively. Expression in equation (14) calculates a new velocity for each particle based on the particle's previous velocity, the particle's location at which the best fitness has been achieved so far, and the population global location at which the best fitness has been achieved so far [11]. In addition, c1 and c2 are positive constants called the cognitive parameter and the social parameter, respectively. These constants provide the correct balance between exploration and exploitation (individuality and sociality). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward p-best and g-best locations. The random numbers provide a stochastic characteristic for the particles velocities in order to simulate the real behavior of the birds in a flock. Figure 1 shows the concept of modification of searching points described by expression in equation (14). An inertia weight parameter w was introduced in order to improve the performance of the original particle swarm optimization model. This parameter plays the role of balancing the global search and local search capability of particle swarm optimization. It can be a positive constant or even a positive linear or non linear function of time.

A better method of global optimum within a reasonable number of iterations can be achieved by incorporating this parameter into the velocity update expression in equation (14), as follows:

$$V_i^{k+1} = wV_i^k + C_1 rand_1(.) \times (p_{best_i} - s_i^k) + C_2 rand_2(.) \times (g_{best_i} - s_i^k) \quad (16)$$

The following weighting function is usually utilized in equation (14)

$$w_i = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter \quad (17)$$

Where,

W_{max} = initial weight, Wmin = final weight, maxIter = maximum iteration number, iter = current iteration number

S^K = Current searching point, S^{K+1} = Modified searching point, V^K = Current velocity, V^{K+1} = Modified velocity

V_{pbest} = Velocity based on p_{best}, V_{gbest} = Velocity based on g_{best}

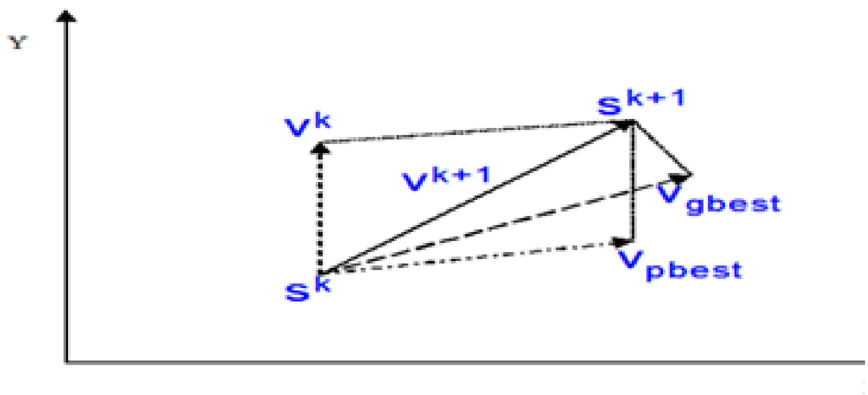

Fig. 1 concept of modification of a searching point by PSO

3.1.1 *PSO Algorithm Procedure:*

Step 1: Generation of initial condition of each agent. Initial searching points (s_i^0) and velocities (v_i^0) of each agent are usually generated randomly within the allowable range. The current searching point is set to pbest for each agent. The best evaluated value of pbest is set to gbest, and the agent number with the best value is stored.

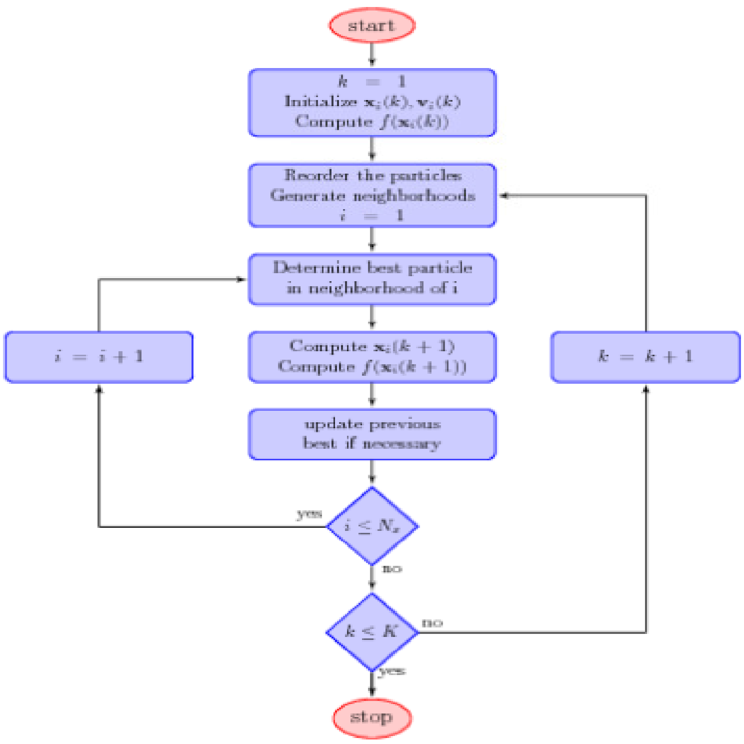

Fig. 2 PSO algorithm

Step 2: Evaluation of searching point of each agent. The objective function value is calculated for each agent. If the value is better than the current pbest of the agent, the pbest

value is replaced by the current value. If the best value of pbest is better than the current gbest, gbest is replaced by the best value and the agent number with the best value is stored.
Step 3: Modification of each searching point.
Step 4: The current iteration number reaches the predetermined maximum iteration number, then exits. Otherwise, the process proceeds to step 2.

3.1.2 Selection of Parameters for PSO

The main parameters of the canonical PSO model are ω, C_1, C_2, $V_{\max}$ and the swarm size *S*. The settings of these parameters determine how it optimizes the search-space. For instance, one can apply a general setting that gives reasonable results on most problems, but seldom is very optimal. Since the same parameter settings not at all guarantee success in different problems, we must have knowledge of the effects of the different settings, such that we can pick a suitable setting from problem to problem.

3.1.3 The Inertia Weight ω

The inertia weight ω controls the momentum of the particle: If $\omega << 1$, only little momentum is preserved from the previous time-step; thus quick changes of direction are possible with this setting. The concept of velocity is completely lost if $\omega = 0$, and the particle then moves in each step without knowledge of the past velocity. On the other hand, if ω is high (>1) we observe the same effect as when C_1 and C_2 are low: Particles can hardly change their direction and turn around, which of course implies a larger area of exploration as well as a reluctance against convergence towards optimum.

3.1.4 The Maximum Velocity V_{max}

The maximum velocity V_{max} determines the maximum change one particle can undergo in its positional coordinates during iteration. Usually we set the full search range of the particle's position as the V_{max}. For example, in case, a particle has position vector $x = (x_1, x_2, x_3)$ and if $-10 \le xi \le 10$ for $i = 1, 2$ and 3, then we set $V_{max} = 20$. Originally, V_{max} was introduced to avoid explosion and divergence. However, with the use of constriction factor χ (to be discussed shortly) or ω in the velocity update formula, $V_{max}$to some degree has become unnecessary; at least convergence can be assured without it [6].

3.1.5 The Acceleration Coefficients C_1 and C_2

A usual choice for the acceleration coefficients C_1 and C_2 is $C_1 = C_2 = 1.494$ [11]. However, other settings were also used in different papers. Usually C_1 equals to C_2 and ranges from (0, 4).

$$C_1 = \left(C_{1f} - C_{1i}\right)\frac{iter}{MAXITER} + C_{1i} \tag{18}$$

$$C_2 = \left(C_{2f} - C_{2i}\right)\frac{iter}{MAXITER} + C_{2i} \tag{19}$$

Where C_{1i}, C_{1f}, C_{2i}, and C_{2f} are constants, iteration is the current iteration number and MAXITER is the number of maximum allowable iterations [12]. The objective of this modification was to boost the global search over the entire search space during the early part of the optimization and to encourage the particles to converge to global optima at the end of the search. Essentially C_1 was decreased from 2.5 to 0.5 whereas C_2 was increased from 0.5 to 2.5 [14].

3.1.6 The Neighborhood Topologies in PSO

The neighborhood of each particle is generally defined as topologically nearest particles to the particle at each side. The global version of PSO also can be considered as a local version of PSO with each particle's neighborhood to be the whole population. It has been suggested that the global version of PSO converges fast, but with potential to converge to the local minimum, while the local version of PSO might have more chances to find better solutions slowly. Since then, a lot of researchers have worked on improving its performance by designing or implementing different types of neighborhood structures in PSO. Kennedy claimed that PSO with small neighborhoods might perform better on complex problems while PSO with large neighborhood would perform better for simple problems [13]. The *k*-best topology, proposed by Kennedy connects every particle to its *k* nearest particles in the topological space. With *k* = 2, this becomes the circle topology (and with *k* = swarmsize-1 it becomes a gbest topology).

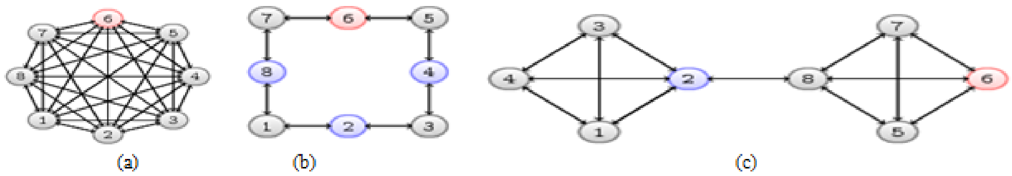

Fig. 3 (a) The fully connected org best topology (b) Ring topology and (c) The wheel topology

4. SIMULATION RESULTS

4.1 Minimization of system power losses (P_{Loss})

In the first case, the proposed algorithm is run with minimization of real power losses as the objective function. The parameters of the proposed method are exposed in table 1. Table 2 presents the result of the study for PL and VD for the IEEE 26 bus test network. The algorithm reaches a minimum loss of **17.3659 MW.**

Table 1 Parameters of the proposed method

Parameters	Values
Number of generations	500
Max. and Min. inertia weights	0.9 and 0.4
Population size	60
Acceleration constants (C1, C2)	2.0 and 2.0

Table 2. Result of the study for P_L and VD (IEEE 26 bus)

Proposed Method	Voltage deviations (p.u.)	Power loss(MW)
Particle swarm optimization	0.3301	17.3659 MW.

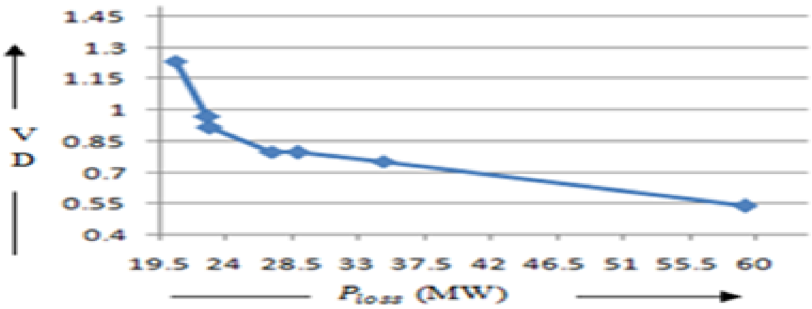

Fig. 4 Pareto optimal between Ploss and VD

Table 3 Result of the study for P_L and VD of the proposed approach (IEEE 26 bus)

Values	Initial setting	Optimal setting of Ploss (MW)	Optimal setting of VD
V_1	1.025	1.0388	1.0319
V_2	1.020	1.0458	1.0300
V_3	1.025	1.0351	1.0122
V_4	1.050	1.0429	1.0325
V_5	1.045	1.0456	1.0599
V_{26}	1.015	1.0361	1.0149
T_3	0.960	1.0080	1.0291
T_6	0.960	1.0735	1.0817
T_8	1.017	1.0042	1.0132
T_9	1.050	1.0160	1.0616
T_{10}	1.050	1.0403	1.0028
T_{15}	0.150	1.0278	1.0028
T_{18}	0.950	1.0106	1.0604

Qc_1	4.000	1.2878	3.7105
Qc_4	2.000	1.0072	1.3779
Qc_5	5.000	0.1314	3.3865
Qc_6	2.000	0.5724	2.9841
Qc_9	3.000	2.2561	3.3343
Qc_{11}	1.500	4.2777	3.4452
Qc_{12}	2.000	4.9781	0.6362
Qc_{15}	0.500	1.1263	3.2656
Qc_{19}	5.000	5.000	0.3588
$P_{loss\ (MW)}$	17.7595	**17.4659**	
VD (p.u.)	0.4311		**0.3012**

This case, the problem is treated as a multi-objective reactive power dispatch problem where both objective functions i.e. real power loss (P_{LOSS}) and voltage deviation (VD) are optimized simultaneously with the proposed PSO algorithm. The diversity of the Pareto optimal set with 7 non-dominated solutions over the trade-off curve is shown in the Figure 4. Pareto-optimal front of proposed approach it is clear that best compromise solution is obtained at weighing factor **w_1=1.0250** and at this weighing factor real power loss and voltage deviations are **17.3659 MW** and **0.3012p.u.**

5. CONCLUSION

In this paper particle swarm optimization algorithm method is applied to getting solves the reactive power dispatch of power system. The proposed algorithm is applied on IEEE 26-bus test system results for minimization of active power loss and voltage profile (VD). The results obtained from the methods proved that they are efficient in solving the reactive power dispatch problem. Other main advantage of particle swarm optimization over other modern heuristics is modeling stability, flexibility, sure and fast convergence than other heuristic methods. Particle swarm optimization method requires only a few parameters to be adjusted, which make easy method to implement and high quality solution. The Pareto graph shows the feasibility of the proposed method for reactive power dispatch is demonstrated on simple power systems with promising results.

References

[1] Kenji Iba, Hiroshi Suzuki, ken-ichi-Suzuki, Katsuhiko Suzuki, "Practical reactive power allocation/operation planning using successive linear programming", IEEE Trans on Power Syst, Vol 3, No2, May 1988, pp 558-566.

[2] Hsiao- Dong Chiang, Jin- Cheng Wang, Orville Cockings and Hyon-Duck Shin "Optimal Capacitor Placements in Distribution Systems: Part 1: A new Formulation and the Overall Problem" IEEE Trans, power Delivery ,Vol 5, No 2,April 1990, pp 634-642.

[3] K. H Abdul Rahaman, S. M. Shahidehpour "Reactive Power Optimization Using Fuzzy Load Representation" IEEE Trans, Power Syst, Vol 9, May 1994, pp 898-90.

[4] B.Gao ,G.K Morison P.Kundur ,'voltage stability evaluation using modal analysis ' Transactions on Power Systems ,Vol 7, No. 4 November 1992.

[5] DOMMEL, H. W.: Optimal Power Dispatch, IEEE Transactions on Power Apparatus and Systems PAS93 No. 3 (May/june 1974), 820–830.

[6] Angeline, P. J. "Using selection to improve particle swarm optimization", Proceedings of IEEE Congress on Evolutionary Computation 1998 Anchorage, Alaska, USA, 1998.

[7] H. T. Yang, P. C. Yang, and C. L. Huang, "Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions", IEEE Trans. on Power Systems, Vol. 11, No. 1, Feb. 1996, pp. 112-118.

[8] Kennedy J, Eberhart R, Shi Y. Swarm intelligence. San Francisco: Morgan Kaufmann Publishers; 2001.

[9] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evolut Comput 2002; 6(1):58–73.

[10] Call JM. Genetic algorithms for modeling and optimization. J Comput Appl Math 2005; 184:205–22.

[11] Wen Zhang and Yutian Liu, "multi-objective reactive power and voltage control based on fuzzy optimization strategy and fuzzy adaptive particle swarm", Electrical power and Energy Systems, 2008.

[12] S. Chu and J. Roddick. A Clustering Algorithm Using Tabu Search Approach with Simulated Annealing for Vector Quantization. Chinese Journal of Electronics, vol. 12, no. 3, 2003, pp. 349-353.

[13] 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.

[14] Vivek Kumar Jain, Himmat Singh, and Laxmi Srivastva "Minimization of Reactive Power Using Particle Swarm Optimization". IJCER, May-June 2012, Vol. 2, Issue No.3, 686-691.