An Improved Great Deluge Algorithm (IGDA) for Solving Optimal Reactive Power Dispatch Problem

K. Lenin, B. Ravindranath Reddy, and M. Surya Kalavathi
Jawaharlal Nehru Technological University Kukatpally, Hyderabad 500 085, India
Email: gklenin@gmail.com

***Abstract*—This paper presents Improved Great Deluge Algorithm (IGDA) for solving the multi-objective reactive power dispatch problem. Modal analysis of the system is used for static voltage stability assessment. Loss minimization and maximization of voltage stability margin are taken as the objectives. Generator terminal voltages, reactive power generation of the capacitor banks and tap changing transformer setting are taken as the optimization variables. Like other local search approaches, this approach also replaces common solution (New_Config) with best results (Best_Config) that have been found by then. This action continues until stop conditions is provided. In this algorithm, new solutions are selected from neighbours. Selection strategy is different from other approaches. In order to evaluate the proposed algorithm, it has been tested on IEEE 30 bus system and compared to other algorithms reported those before in literature. Results show that IGDA is more efficient than others for solution of single-objective ORPD problem.**

***Index Terms*—modal analysis, optimal reactive power, transmission loss, improved great deluge algorithm (IGDA), optimization**

I. Introduction

Optimal reactive power dispatch problem is one of the difficult optimization problems in power systems. The sources of the reactive power are the generators, synchronous condensers, capacitors, static compensators and tap changing transformers. The problem that has to be solved in a reactive power optimization is to determine the required reactive generation at various locations so as to optimize the objective function. Here the reactive power dispatch problem involves best utilization of the existing generator bus voltage magnitudes, transformer tap setting and the output of reactive power sources so as to minimize the loss and to enhance the voltage stability of the system. It involves a non linear optimization problem. Various mathematical techniques have been adopted to solve this optimal reactive power dispatch problem. These include the gradient method [1]-[2], Newton method [3] and linear programming [4]-[7].The gradient and Newton methods suffer from the difficulty in handling inequality constraints. To apply linear programming, the input- output function is to be expressed as a set of linear functions which may lead to loss of accuracy. Recently global optimization techniques such as genetic algorithms have been proposed to solve the reactive power flow problem [8]-[11]. In recent years, the problem of voltage stability and voltage collapse has become a major concern in power system planning and operation. To enhance the voltage stability, voltage magnitudes alone will not be a reliable indicator of how far an operating point is from the collapse point [12]. The reactive power support and voltage problems are intrinsically related. Hence, this paper formulates the reactive power dispatch as a multi-objective optimization problem with loss minimization and maximization of static voltage stability margin (SVSM) as the objectives. Voltage stability evaluation using modal analysis [12] is used as the indicator of voltage stability. The Great Deluge algorithm (GD) [13] is a generic algorithm applied to optimization problems. It is similar in many ways to the hill-climbing and simulated annealing algorithms. The name comes from the analogy that in a great deluge a person climbing a hill will try to move in any direction that does not get his/her feet wet in the hope of finding a way up as the water level rises. In a typical implementation of the GD, the algorithm starts with a poor approximation, S, of the optimum solution. A numerical value called the badness is computed based on S and it measures how undesirable the initial approximation is. The higher the value of badness the more undesirable is the approximate solution. Another numerical value called the tolerance is calculated based on a number of factors, often including the initial badness.

A new approximate solution S', called a neighbour of S, is calculated based on S. The badness of S', b', is computed and compared with the tolerance. If b' is better than tolerance, then the algorithm is recursively restarted with S = S', and tolerance = decay (tolerance), where decay is a function that lowers the tolerance (representing a rise in water levels). If b' is worse than tolerance, a different neighbour S* of S is chosen and the process repeated. If all the neighbours of S produce approximate solutions beyond tolerance, then the algorithm is terminated and S is put forward as the best approximate solution obtained. In this paper a new model known as improved Great Deluge algorithm (IGDA) is proposed for

Manuscript received October 15, 2013; revised March 7, 2014.

doi: 10.12720/ijeee.2.4.321-326

solving reactive power optimization problem. In this work, we utilize a Great Deluge (GD) algorithm that was introduced by Dueck [13] and applied by Burke *et al.* [14] in different optimization problem. Then, enhance GD algorithm by proposing an improved Great Deluge algorithm to overcome some of the limitation of GD. In proposed model, global and local characters of the algorithms are used in an efficient way.

II. VOLTAGE STABILITY EVALUATION

A. Modal Analysis for Voltage Stability Evaluation

Modal analysis is one of the methods for voltage stability enhancement in power systems. The linearized steady state system power flow equations are given by.

$$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = \begin{bmatrix} J_{p\theta} & J_{pv} \\ J_{q\theta} & J_{QV}\end{bmatrix} \quad (1)$$

where

ΔP = Incremental change in bus real power.
ΔQ = Incremental change in bus reactive Power injection
$\Delta\theta$ = incremental change in bus voltage angle.
ΔV = Incremental change in bus voltage Magnitude

$J_{p\theta}$, J_{PV}, $J_{Q\theta}$, J_{QV} jacobian matrix are the sub-matrixes of the System voltage stability is affected by both P and Q. However at each operating point we keep P constant and evaluate voltage stability by considering incremental relationship between Q and V.

To reduce (1), let $\Delta P = 0$, then.

$$\Delta Q=[J_{QV} - J_{Q\theta} J_{P\theta}^{-1} J_{PV}]\Delta V = J_R \Delta V \quad (2)$$

$$\Delta V = J^{-1}\Delta Q \quad (3)$$

where

$$J_R= (J_{QV} - J_{Q\theta} J_{P\theta}^{-1} J PV) \quad (4)$$

J_R is called the reduced Jacobian matrix of the system.

B. Modes of Voltage Instability

Voltage Stability characteristics of the system can be identified by computing the eigen values and eigen vectors

Let

$$J_R = \xi\wedge\eta \quad (5)$$

where,

ξ = right eigenvector matrix of J_R

η = left eigenvector matrix of J_R

$\wedge$ = diagonal eigenvalue matrix of J_R and

$$J_{R^{-1}} = \xi\wedge^{-1}\eta \quad (6)$$

From Eq. (3) and Eq. (6), we have

$$\Delta V = \xi\wedge^{-1}\eta\Delta Q \quad (7)$$

Or

$$\Delta V = \sum_I \frac{\xi_i\eta_i}{\lambda_i}\Delta Q \quad (8)$$

where ξ_i is the ith column right eigenvector and η the ith row left eigenvector of J_R. λ_i is the ith eigen value of J_R.

The ith modal reactive power variation is,

$$\Delta Q_{mi} = K_i\xi_i \quad (9)$$

where,

$$K_i = \sum_j \xi_{ij^2} - 1 \quad (10)$$

where ξ_{ji} is the jth element of ξ_i

The corresponding ith modal voltage variation is

$$\Delta V_{mi} = [1/\lambda_i]\Delta Q_{mi} \quad (11)$$

In (8), let $\Delta Q = e_k$, where e_k has all its elements zero except the kth one being 1. Then,

$$\Delta V = \sum_i \frac{\eta_{1k}\,\xi_1}{\lambda_1} \quad (12)$$

η_{1k} k th element of η_1

V –Q sensitivity at bus k

$$\frac{\partial V_K}{\partial Q_K} = \sum_i \frac{\eta_{1k}\,\xi_1}{\lambda_1} = \sum_i \frac{P_{ki}}{\lambda_1} \quad (13)$$

III. PROBLEM FORMULATION

The main objective of the reactive power dispatch problem is to minimize the system real power loss and maximize the static voltage stability index margins .

A. Minimization of Real Power Loss

Minimization of real power loss (Ploss) in transmission lines of a power system is mathematically stated as follows.

$$P_{loss}= \sum_{\substack{k=1\\ k=(i,j)}}^{n} g_{k(V_i^2+V_j^2-2V_i V_j \cos\theta_{ij})} \quad (14)$$

where n is the number of transmission lines, g_k is the conductance of branch k, V_i and V_j are voltage magnitude at bus i and bus j, and θij is the voltage angle difference between bus i and bus j.

B. Minimization of Voltage Deviation

The minimization of the Deviations in voltage magnitudes (VD) at load buses is mathematically stated as follows.

$$\text{Minimize VD} = \sum_{k=1}^{nl}|V_k - 1.0| \quad (15)$$

where nl is the number of load busses and V_k is the voltage magnitude at bus k.

C. System Constraints

Objective functions are subjected to these constraints shown below.

Load flow equality constraints:

$$P_{Gi} - P_{Di} - V_i \sum_{j=1}^{nb} V_j \begin{bmatrix} G_{ij} & \cos\theta_{ij} \\ +B_{ij} & \sin\theta_{ij}\end{bmatrix} = 0, i = 1,2 \ldots, nb \quad (16)$$

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=1}^{nb} V_j \begin{bmatrix} G_{ij} & \cos\theta_{ij} \\ +B_{ij} & \sin\theta_{ij} \end{bmatrix} = 0, i = 1,2 \ldots., nb \quad (17)$$

where, nb is the number of buses, P_G and Q_G are the real and reactive power of the generator, P_D and Q_D are the real and reactive load of the generator, and G_{ij} and B_{ij} are the mutual conductance and susceptance between bus i and bus j.

Generator bus voltage (V_{Gi}) inequality constraint:

$$V_{Gi}^{min} \leq V_{Gi} \leq V_{Gi}^{max}, i \in ng \quad (18)$$

Load bus voltage (V_{Li}) inequality constraint:

$$V_{Li}^{min} \leq V_{Li} \leq V_{Li}^{max}, i \in nl \quad (19)$$

Switchable reactive power compensations ($Q_C i$) inequality constraint:

$$Q_{Ci}^{min} \leq Q_{Ci} \leq Q_{Ci}^{max}, i \in nc \quad (20)$$

Reactive power generation (Q_{Gi}) inequality constraint:

$$Q_{Gi}^{min} \leq Q_{Gi} \leq Q_{Gi}^{max}, i \in ng \quad (21)$$

Transformers tap setting (T_i) inequality constraint:

$$T_i^{min} \leq T_i \leq T_i^{max}, i \in nt \quad (22)$$

Transmission line flow (S_{Li}) inequality constraint:

$$S_{Li}^{min} \leq S_{Li}^{max}, i \in nl \quad (23)$$

where, nc, ng and nt are numbers of the switchable reactive power sources, generators and transformers

IV. Great Deluge Algorithm

Great deluge algorithm is a comprehensive approach for solving optimization problems which first Dueck suggested in 1993. Like other local search approaches, this approach also replaces common solution (*New_Config*) with best results (*Best_Config*) that have been found by then. This action continues until stop conditions is provided. In this algorithm, new solutions are selected from neighbours. Selection strategy is different from other approaches. In great deluge algorithm these results are acceptable which their values are equal or better than (for optimization problems) the value of *Water Level* (*WL)*. Value of *WL* also rises at a steady pace in every step. Increase of *WL* continues until value of *WL* equals with the best result achieved ever. In this step, the algorithm is repeated several times and if better result is not obtained, it comes to the end. The primary amount of *WL* is equal with the primary results (*f(s)*).

V. Improved Great Deluge Algorithm

The great deluge algorithm was introduced by Dueck [13]. It is a local search procedure which has certain similarities with other (i.e. simulated annealing, SA) algorithms, but less dependent upon parameters, where it is just two parameters such as the amount of computational time and an estimate of the quality of solution. Great deluge always accepts an improve solution and a worse solution is accepted if the quality of the solution is less than (for the case or minimisation) or equal to an upper boundary or ―level‖. During the search process, the ―level‖ is iteratively updated by a constant decreasing rate β. In GD, the candidate (new) solution can be accepted if its quality (minimal distance) is better than the best solution (*SArrange*) quality, or accepts a little worse solution if it is better than the *level* (acceptance criterion). Then the *level* is decreased by β, where β is calculated by using the formula, adopted from Burke et al. [14] (which is used in different optimization problem) as shows in Equation (24). This process will be repeated until the stopping condition is met.

$$\beta = (f(So) - est.q) / N.iters \quad (24)$$

The Great Deluge algorithm (GD) starts with a given *K-Means* partitions i.e. the initial solution is generated by *K-Means* algorithm. Again we list the notations used in this work below:

- *So*: initial solution
- *f*(*So*): quality of *So*
- *SA*rrange: best solution
- f(SArr*ange*): the quality of *SArrange*
- *Ssour*ce: the current solution
- f(Ssour*ce*): the quality of *Ssource*
- *Sworki*ng: the candidate solution
- f(Sworki*ng*): the quality of *S working*
- *level*: boundary
- est.q: estimated quality of the final solution
- *N.iters*: number of iterations
- Iterations: iteration counter
- *β*: decreasing rate
- not_impro*ving_length_GD* : maximum number of iterations where there is not improvement in the quality of the solution

In this work, at the beginning of the search, the *level* is set to be initial water level. The water level, *level*, is decreased by *β* in each of the iteration where *β* is based on the estimated quality (*est.q*). The pseudo code for the GD to solve clustering problems is shown in Fig. 1. Fig. 1 shows that, the algorithm starts by initializing the required parameters as in Step-1 by setting the stopping condition (*N.iters*), estimated quality of the final solution (*est.q*), the initial water level (*level*), decreasing rate (*β*), maximum number of not improving solutions (*not_improving_length_GD)*. Again, note that the initial solution is generated using *K-Means* (*So*).

In the improvement phase (Step-2), neighbourhood structures N1 and N2 are applied to generate candidate solutions (in this case, five candidate solutions are generated), and the best candidate is selected as the candidate solution (*Sworking*) as shown in Step-2.1. In this work there are two cases to be taken into consideration as follows:

- Case 1: Better solution
  If *f*(*Sworking*) is better than *f*(*SArrange*), then *Sworking* is accepted as a current solution (*Ssource* ← *Sworking*), and the best solution is

updated (*SArrange* ← *Sworking*) as shown in Step-2.2. The *level* will be updated by the value β (i.e. *level* = *level* - β).

- Case 2: Worse solution

  If *f*(*Sworking*) is less than *f*(*SArrange*), then the quality of *Sworking* is compared against the level. If it is less than or equal to the *level*, then *Sworking* is accepted, and the current solution is updated (*Ssource* ← *Sworking*). Otherwise, *Sworking* will be rejected. The *level* will be updated by the value β (i.e. *level* = *level* - β). The counter for the non improving solution is increased by 1. If this counter is equal *non_improving_length_GD*, then the process terminates. Otherwise, the process continues the stopping condition is met (i.e. *Iterations*> *N.iters*), and return the best solution found *SArrange*. (Step-2*)*. Note that in this work the *est.q* is set to 0, and *non_improving_length_GD* is set to 10 (after some preliminary experiments).

Step-1: Initialization Phase
Determine initial candidate solution So and f(So);
SArrange = So; f(SArrange)= f(So);
Ssource = So; f(Ssource)= f(So);
Set N.iters; (stopping condition)
Set estimated quality of final solution, est.q;
Set not_improving_length_GD;//maximum number of GD not improved
level= f(So); // initial level
decreasing rate β = ((f(So) - est.q) / (N.iters));
Iterations=0; not_improving_counter=0;
Step-2: Improvement (Iterative) Phase
repeat (*while termination condition is not satisfied)*
Step-2.1: *Selecting candidate solution Sworking*
Generate candidate solutions by applying neighbourhood structure
(N1 and N2) and the best solution consider as candidate
solution (Sworking);
Step-2.2: Accepting Solution
if *f(Sworking) < f(SArrange)*
SArrange = Sworking; f(SArrange)=f(Sworking);
Ssource = Sworking; f(Ssource)=f(Sworking);
not_improving_counter = 0;
else
if *f (Sworking) ≤ level*
Ssource = Sworking;
else
Increase not_improving_counter by one;
if not_improving_counter
==not_improving_length_GD,
exit;
end if
level = level - β;
end if
Iterations= Iterations+1;
until *Iterations > N.iters (termination condition is met)*
Step-3: Termination phase
Return the best found solution SArrange

Figure 1. Code for great deluge algorithm

Step-1: Initialization Phase

Determine initial candidate solution So and f(So);

SArrange = So; f(SArrange)= f(So); Ssource = So; f(Ssource)= f(So);

Set N.iters; (stopping condition)

Set estimated quality of final solution, est.q;

Set not_improving_length_GD;//maximum number of GD not improved

level= f(So); // initial level

Initialize all element in MGD list (LMGD) = Level;

Set Lsize ; CountrMGD =0; // MGD

decreasing rate β = ((f(So) - est.q) / (N.iters));

Iterations=0; not_improving_counter=0;

Step-2: Improvement (Iterative) Phase

repeat (*while termination condition is not satisfied)*

Step-2.1: *Selecting candidate solution Sworking*

Generate candidate solutions by applying neighbourhood structure (N1 and N2) and the best solution consider as candidate solution (Sworking);

Step-2.2: Accepting Solution

if *f(Sworking) < f(SArrange)*

SArrange = Sworking; f(SArrange)=f(Sworking);

Ssource = Sworking; f(Ssource)=f(Sworking);

not_improving_counter = 0;

CountrMGD = CountrMGD +1; // MGD

IndexMGD = CountrMGD mod Lsize; // MGD

LMGD (IndexMGD) = level; // MGD

else

if *f (Sworking) ≤ level*

Ssource = Sworking;

else

Increase not_improving_counter by one;

if not_improving_counter ==not_improving_length_GD,

RN= random number between 1 and Lsize; // MGD

level = LMGD (RN) // MGD

end if

level = level - β;

end if

Iterations= Iterations+1;

until *Iterations > N.iters (termination condition are met)*

Step-3: Termination phase

Return the best found solution SArrange

Figure 2. Code for modified great deluge algorithm

However, there are three drawbacks in employing the GD algorithm over clustering problems such as: (i) in GD the estimated quality (*est.q*) of the final solution is very hard to investigate, as each dataset has it is own

performance (e.g. the solution in some datasets is improved with big differences and in some datasets the solution is improved with a different range), (ii) in GD the acceptance criterion is based on level which is decreased based on the estimated quality (see Equation (3.2)) that is decreased continuously without control, and (iii) in GD the neighbourhood structure i.e. N1 and N2 are not really effective as it is based at random. Therefore, the Improved Great Deluge Algorithm (IGDA) is proposed to overcome these drawbacks. IGD structure resembles the original structure of the GD algorithm, but the basic difference is in term of updating the *level*. In MGD, we have introduce a list that keeps the previous *level* value at the time when the better solution is obtained (i.e. *SArrange* = *Sworking*). When the maximum number of iteration of no improved GD (*not_improving_length_GD*) is met, then the *level* is updated by a new *level* that is randomly selected from the list (where the size of the list is set to 10 based on preliminary experiments). The pseudo code for the MGD algorithm is presented in Fig. 2.

VI. Simulation Results

The validity of the proposed Algorithm technique is demonstrated on IEEE-30 bus system. The IEEE-30 bus system has 6 generator buses, 24 load buses and 41 transmission lines of which four branches are (6-9), (6-10), (4-12) and (28-27) - are with the tap setting transformers. The real power settings are taken from [1]. The lower voltage magnitude limits at all buses are 0.95p.u. and the upper limits are 1.1 for all the PV buses and 1.05p.u. for all the PQ buses and the reference bus.

TABLE I. Voltage Stability under Contingency State

Sl. No	Contigency	ORPD Setting	Vscrpd Setting
1	28-27	0.1400	0.1422
2	4-12	0.1658	0.1662
3	1-3	0.1784	0.1754
4	2-4	0.2012	0.2032

TABLE II. Limit Violation Checking Of State Variables

State variables	limits		ORPD	VSCRPD
	Lower	upper		
Q1	-20	152	1.3422	-1.3269
Q2	-20	61	8.9900	9.8232
Q5	-15	49.92	25.920	26.001
Q8	-10	63.52	38.8200	40.802
Q11	-15	42	2.9300	5.002
Q13	-15	48	8.1025	6.033
V3	0.95	1.05	1.0372	1.0392
V4	0.95	1.05	1.0307	1.0328
V6	0.95	1.05	1.0282	1.0298
V7	0.95	1.05	1.0101	1.0152
V9	0.95	1.05	1.0462	1.0412
V10	0.95	1.05	1.0482	1.0498
V12	0.95	1.05	1.0400	1.0466
V14	0.95	1.05	1.0474	1.0443
V15	0.95	1.05	1.0457	1.0413
V16	0.95	1.05	1.0426	1.0405
V17	0.95	1.05	1.0382	1.0396
V18	0.95	1.05	1.0392	1.0400
V19	0.95	1.05	1.0381	1.0394
V20	0.95	1.05	1.0112	1.0194
V21	0.95	1.05	1.0435	1.0243
V22	0.95	1.05	1.0448	1.0396
V23	0.95	1.05	1.0472	1.0372
V24	0.95	1.05	1.0484	1.0372
V25	0.95	1.05	1.0142	1.0192
V26	0.95	1.05	1.0494	1.0422
V27	0.95	1.05	1.0472	1.0452
V28	0.95	1.05	1.0243	1.0283
V29	0.95	1.05	1.0439	1.0419
V30	0.95	1.05	1.0418	1.0397

TABLE III. Comparison of Real Power Loss

Method	Minimum loss
Evolutionary programming[15]	5.0159
Genetic algorithm[16]	4.665
Real coded GA with Lindex as SVSM[17]	4.568
Real coded genetic algorithm[18]	4.5015
Proposed IGDA method	4.4322

VII. Conclusion

In this paper a novel approach IGDA algorithm used to solve optimal reactive power dispatch problem.The proposed method formulates reactive power dispatch problem as a mixed integer non-linear optimization problem and determines control strategy with continuous and discrete control variables such as generator bus voltage, reactive power generation of capacitor banks and on load tap changing transformer tap position. To handle the mixed variables a flexible representation scheme was proposed. The performance of the proposed algorithm demonstrated through its voltage stability assessment by modal analysis is effective at various instants following system contingencies. Also this method has a good performance for voltage stability Enhancement of large, complex power system networks. The effectiveness of the proposed method is demonstrated on IEEE 30-bus system.

References

[1] O. Alsac and B. Scott, "Optimal load flow with steady state security," *IEEE Transaction*, pp. 745-751, 1973.

[2] K. Y. Lee, Y. M. Paru, and J. L. Oritz, "A united approach to optimal real and reactive power dispatch," *IEEE Transactions on Power Apparatus and Systems*, pp. 1147-1153, 1985.

[3] A. Monticelli, M. V. F. Pereira, and S. Granville, "Security constrained optimal power flow with post contingency corrective rescheduling," *IEEE Transactions on Power Systems*, vol. 2, no. 1, pp. 175-182, 1987.

[4] N. Deeb and S. M. Shahidehpur, "Linear reactive power optimization in a large power network using the decomposition approach," *IEEE Transactions on Power System*, vol. 5, no. 2, pp. 428-435, 1990

[5] E. Hobson, "Network consrained reactive power control using linear programming," *IEEE Transactions on Power Systems*, pp. 868-877, 1980

[6] K. Y. Lee, Y. M. Park, and J. L. Oritz, "Fuel–cost optimization for both real and reactive power dispatches," *Generation,*

Transmission and Distribution, IEE Proceedings C, vol. 131, no. 3, pp. 85-93, 1984.
[7] M. K. Mangoli and K. Y. Lee, "Optimal real and reactive power control using linear programming," *Electr. Power Syst. Res.*, vol. 26, pp. 1-10, 1993.
[8] S. R. Paranjothi and K. Anburaja, "Optimal power flow using refined genetic algorithm," *Electr. Power Compon. Syst.*, vol. 30, pp. 1055-1063, 2002.
[9] D. Devaraj and B. Yeganarayana, "Genetic algorithm based optimal power flow for security enhancement," *IEE Proc.-Generation, Transmission and Distribution*, vol. 152, no. 6, pp. 899-905, 2005.
[10] C. J. A. B. Filho, F. B. de Lima Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima, *Fish School Search Nature – Inspired Algorithms for Optimization Studies in Computational Intelligence*, vol. 193, 2009, pp. 261-277.
[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, IEEE Service Center, Piscataway, NJ, 1995, pp. 1942-1948.
[12] C. A. Canizares, A. C. Z. de Souza, and V. H. Quintana, "Comparison of performance indices for detection of proximity to voltage collapse," *IEEE Transactions on Power Systems*, vol. 11. no. 3, pp. 1441-1450, Aug. 1996.
[13] G. Dueck, "New optimization heuristics: Great deluge and the record-to-record travel," *Journal of Computational Physics*, vol. 104, no. 1, pp. 86-92, 1993.
[14] E. K. Burke, Y. Bykov, J. P. Newall, and S. Petrovic, "A time-predefined local search approach to exam timetabling problem," *IEEE Transactions*, vol. 36, no. 6, pp. 509-528, June 2004.
[15] Q. H. Wu and J. T. Ma, "Power system optimal reactive power dispatch using evolutionary programming." *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1243-1248, 1995.
[16] S. Durairaj, D. Devaraj, and P. S. Kannan, "Genetic algorithm applications to optimal reactive power dispatch with voltage stability enhancement," *IE (I) Journal-EL*, vol. 87, Sep. 2006.
[17] D. Devaraj, "Improved genetic algorithm for multi–objective reactive power dispatch problem," *European Transactions on Electrical Power*, vol. 17, pp. 569-581, 2007.
[18] P. A. Jeyanthy and D. Devaraj, "Optimal reactive power dispatch for voltage stability enhancement using real coded genetic algorithm," *International Journal of Computer and Electrical Engineering*, vol. 2, no. 4, pp. 1793-8163, Aug. 2010.

K. Lenin has received his B.E., Degree, electrical and electronics engineering in 1999 from university of madras, Chennai, India and M.E., Degree in power systems in 2000 from Annamalai University, TamilNadu, India. At present pursuing Ph.D., degree at JNTU, Hyderabad, India.

Bhumanapally. Ravindhranath Reddy, Born on 3rd September,1969. Got his B.Tech in Electrical & Electronics Engineering from the J.N.T.U. College of Engg., Anantapur in the year 1991. Completed his M.Tech in Energy Systems in IPGSR of J. N. T. University Hyderabad in the year 1997. Obtained his doctoral degree from JNTUA, Anantapur University in the field of Electrical Power Systems. Published 12 Research Papers and presently guiding 6 Ph.D. Scholars. He was specialized in Power Systems, High Voltage Engineering and Control Systems. His research interests include Simulation studies on Transients of different power system equipment.

M. Surya Kalavathi has received her B.Tech. Electrical and Electronics Engineering from SVU, Andhra Pradesh, India and M.Tech, power system operation and control from SVU, Andhra Pradesh, India. She received her Ph.D. Degree from JNTU, hyderabad and Post doc. From CMU – USA. Currently she is Professor and Head of the electrical and electronics engineering department in JNTU, Hyderabad, India and she has Published 16 Research Papers and presently guiding 5 Ph.D. Scholars. She has specialised in Power Systems, High Voltage Engineering and Control Systems. Her research interests include Simulation studies on Transients of different power system equipment. She has 18 years of experience. She has invited for various lectures in institutes.