ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: https://doi.org/10.29121/granthaalayah.v5.i8.2017.2215

INTERNATIONAL JOURNAL OF RESEARCH – GRANTHAALAYAH
A knowledge Repository

Science

MINIMIZATION OF REAL POWER LOSS BY ENHANCED GREAT DELUGE ALGORITHM

Dr.K.Lenin [*1]
[*1] Professor, Department of EEE, Prasad V.Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada, Andhra Pradesh -520007, India

Abstract

This paper presents Enhanced Great Deluge Algorithm (EDA) for solving reactive power problem. Alike other local exploration methods, this Enhanced Great Deluge Algorithm (EDA) also swap general solution (fresh_Config) with most excellent results (most excellent_Config) that have been found by then. This deed prolong until stop conditions is offered. In this algorithm, new solutions are selected from neighbours. Selection strategy is different from other approaches. In order to evaluate validity of the proposed Enhanced Great Deluge Algorithm (EDA) algorithm, it has been tested on standard IEEE 118 & practical 191 bus test systems and compared to other standard reported algorithms. Results show that Enhanced Great Deluge Algorithm (EDA) reduces the real power loss and voltage profiles are within the limits.

Keywords: Optimal Reactive Power; Transmission Loss; Enhanced Great Deluge Algorithm; Optimization.

Cite This Article: Dr.K.Lenin. (2017). “MINIMIZATION OF REAL POWER LOSS BY ENHANCED GREAT DELUGE ALGORITHM.” *International Journal of Research - Granthaalayah*, 5(8), 207-216. https://doi.org/10.29121/granthaalayah.v5.i8.2017.2215.

1. Introduction

Optimal reactive power problem is to minimize the real power loss and bus voltage deviation. Various numerical methods like the gradient method [1-2], Newton method [3] and linear programming [4-7] have been adopted to solve the optimal reactive power dispatch problem. Both the gradient and Newton methods have the complexity in managing inequality constraints. If linear programming is applied then the input- output function has to be uttered as a set of linear functions which mostly lead to loss of accuracy. The problem of voltage stability and collapse play a major role in power system planning and operation [8]. Evolutionary algorithms such as genetic algorithm have been already proposed to solve the reactive power flow problem [9-11]. Evolutionary algorithm is a heuristic approach used for minimization problems by utilizing nonlinear and non-differentiable continuous space functions. In [12], Hybrid differential evolution algorithm is proposed to improve the voltage stability index. In [13] Biogeography Based algorithm is projected to solve the reactive power dispatch problem. In [14], a fuzzy based method is used to solve the optimal reactive power scheduling method. In [15], an improved evolutionary programming is used to solve the optimal reactive power dispatch problem. In [16], the optimal reactive power flow problem is solved by integrating a genetic algorithm with a nonlinear interior point method. In [17], a pattern algorithm is used to solve ac-dc optimal reactive power flow model with the generator capability limits. In [18], F. Capitanescu proposes a two-step approach to evaluate Reactive power reserves with respect to operating constraints and voltage stability. In [19], a programming based approach is used to solve the optimal reactive power dispatch problem. In [20], A. Kargarian et al present a probabilistic algorithm for optimal reactive power provision in hybrid electricity markets with uncertain loads. The Great Deluge algorithm (GD) [21] is a generic algorithm and it alike to the hill-climbing and simulated annealing algorithms. The name comes from the resemblance of a great deluge a person climbing a hill will try to move in any direction that does not get his or her feet wet in the anticipate to find a way up as the water level increases. In this work, we utilize a Great Deluge (GD) algorithm that was introduced by Dueck [21] and applied by Burke *et al.* [22] in different optimization problem to solve the reactive power problem. In proposed Enhanced Great Deluge Algorithm (EDA) model, global and local characters of the algorithms are used in a competent way. In order to evaluate validity of the proposed Enhanced Great Deluge Algorithm (EDA) algorithm, it has been tested on standard IEEE 118 & practical 191 bus test systems and compared to other standard reported algorithms. Results show that Enhanced Great Deluge Algorithm (EDA) reduces the real power loss and voltage profiles are within the limits.

2. Problem Formulation

The optimal power flow problem is treated as a general minimization problem with constraints, and can be mathematically written in the following form:

Minimize f(x, u) (1)

subject to g(x,u)=0 (2)
and
$$h(x, u) \leq 0 \quad (3)$$

where f(x,u) is the objective function. g(x.u) and h(x,u) are respectively the set of equality and inequality constraints. x is the vector of state variables, and u is the vector of control variables.
The state variables are the load buses (PQ buses) voltages, angles, the generator reactive powers and the slack active generator power:

$$x = \left(P_{g1}, \theta_2, \ldots, \theta_N, V_{L1}, ., V_{LNL}, Q_{g1}, \ldots, Q_{gng}\right)^T \quad (4)$$

The control variables are the generator bus voltages, the shunt capacitors/reactors and the transformers tap-settings:

$$u = \left(V_g, T, Q_c\right)^T \quad (5)$$

or

$$u = \left(V_{g1}, \ldots, V_{gng}, T_1, \ldots, T_{Nt}, Q_{c1}, \ldots, Q_{cNc}\right)^T \quad (6)$$

Where ng, nt and nc are the number of generators, number of tap transformers and the number of shunt compensators respectively.

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.886353

3. Objective Function

3.1. Active Power Loss

The objective of the reactive power dispatch is to minimize the active power loss in the transmission network, which can be described as follows:

$$F = PL = \sum_{k \in Nbr} g_k \left(V_i^2 + V_j^2 - 2V_i V_j cos\theta_{ij}\right) \quad (7)$$

or

$$F = PL = \sum_{i \in Ng} P_{gi} - P_d = P_{gslack} + \sum_{i \neq slack}^{Ng} P_{gi} - P_d \quad (8)$$

where g_k : is the conductance of branch between nodes i and j, Nbr: is the total number of transmission lines in power systems. P_d: is the total active power demand, P_{gi}: is the generator active power of unit i, and P_{gsalck}: is the generator active power of slack bus.

Voltage Profile Improvement

For minimizing the voltage deviation in PQ buses, the objective function becomes:

$$F = PL + \omega_v \times VD \quad (9)$$

where ω_v: is a weighting factor of voltage deviation.

VD is the voltage deviation given by:

$$VD = \sum_{i=1}^{Npq} |V_i - 1| \quad (10)$$

Equality Constraint

The equality constraint g(x,u) of the ORPD problem is represented by the power balance equation, where the total power generation must cover the total power demand and the power losses:

$$P_G = P_D + P_L \quad (11)$$

This equation is solved by running Newton Raphson load flow method, by calculating the active power of slack bus to determine active power loss.

Inequality Constraints

The inequality constraints h(x,u) reflect the limits on components in the power system as well as the limits created to ensure system security. Upper and lower bounds on the active power of slack bus, and reactive power of generators:

$$P_{gslack}^{min} \leq P_{gslack} \leq P_{gslack}^{max} \quad (12)$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max}, i \in N_g \quad (13)$$

Upper and lower bounds on the bus voltage magnitudes:

$$V_i^{min} \leq V_i \leq V_i^{max}, i \in N \tag{14}$$

Upper and lower bounds on the transformers tap ratios:

$$T_i^{min} \leq T_i \leq T_i^{max}, i \in N_T \tag{15}$$

Upper and lower bounds on the compensators reactive powers:

$$Q_c^{min} \leq Q_c \leq Q_C^{max}, i \in N_C \tag{16}$$

Where N is the total number of buses, N_T is the total number of Transformers; N_c is the total number of shunt reactive compensators.

4. Enhanced Great Deluge Algorithm

Great deluge algorithm replaces common solution (*fresh_Config*) with best results (*most excellent_Config*) that have been found by then. This action continues until stop conditions is offered. In this algorithm, novel solutions are chosen from neighbours. In great deluge algorithm these results are satisfactory which their values are equal or better than the value of *Water Level* (*WL)*. Value of *WL* also increases at a fixed pace in each step. Augment of *WL* persist until value of *WL* equals with the finest result accomplished 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)*).

$$\beta = (f\,(So) – est.q) / N.iters \tag{17}$$

The Great Deluge algorithm 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*
*SAr*range: best solution
f(SArr*ange*): the quality of *SArrange*
*Ssourc*e: 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.

Algorithm of 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
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;
```

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.886353

```
else
if $f\ (Sworking) \le level$
$Ssource = Sworking$;
else
Increase not_improving_counter by one;
if not_improving_counter ==not_improving_length_GD,
exit;
end if
level = level - $\beta$;
end if
Iterations= Iterations+1;
until Iterations > N.iters (termination condition is met)
Step-3: Termination phase
Return the best found solution SArrange
```

However, there are three drawbacks in employing the GD algorithm are : (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 (ii) in GD the acceptance criterion is based on level which is decreased based on the estimated quality 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, Enhanced Great Deluge Algorithm (EDA) is proposed to overcome these drawbacks. EDA 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 arbitrarily selected.

Enhanced great deluge algorithm for solving reactive power problem

```
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 $\beta = ( ( f(So) - est.q ) / (N.iters) )$;
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 $f(Sworking) < f(SArrange)$
```

DOI: 10.5281/zenodo.886353

```
        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) $\le$ 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 - $\beta$;
    end if
    Iterations= Iterations+1;
until Iterations > N.iters (termination condition are met)
Step-3: Termination phase
Return the best found solution SArrange.
```

5. Simulation Results

At first Enhanced Great Deluge Algorithm (EDA) algorithm has been tested in standard IEEE 118-bus test system [23].The system has 54 generator buses, 64 load buses, 186 branches and 9 of them are with the tap setting transformers. The limits of voltage on generator buses are 0.95 - 1.1 per-unit., and on load buses are 0.95 -1.05 per-unit. The limit of transformer rate is 0.9 -1.1, with the changes step of 0.025. The limitations of reactive power source are listed in Table 1, with the change in step of 0.01.

Table 1: Limitation of reactive power sources

BUS	5	34	37	44	45	46	48
QCMAX	0	14	0	10	10	10	15
QCMIN	-40	0	-25	0	0	0	0
BUS	74	79	82	83	105	107	110
QCMAX	12	20	20	10	20	6	6
QCMIN	0	0	0	0	0	0	0

The statistical comparison results of 50 trial runs have been list in Table 2 and the results clearly show the better performance of proposed Enhanced Great Deluge Algorithm (EDA) algorithm.

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.886353

Table 2: Comparison results

Active power loss (p.u)	**BBO [24]**	**ILSBBO/strategy1 [24]**	**ILSBBO/strategy1 [24]**	**Proposed EDA**
Min	128.77	126.98	124.78	116.02
Max	132.64	137.34	132.39	119.24
Average	130.21	130.37	129.22	117.78

Then the Enhanced Great Deluge Algorithm (EDA) algorithm has been tested in practical 191 test system and the following results have been obtained. In Practical 191 test bus system – Number of Generators = 20, Number of lines = 200, Number of buses = 191 Number of transmission lines = 55. Table 3 shows the optimal control values of practical 191 test system obtained by Enhanced Great Deluge Algorithm (EDA) algorithm. And table 4 shows the results about the value of the real power loss by obtained by Enhanced Great Deluge Algorithm (EDA).

Table 3: Optimal Control values of Practical 191 utility (Indian) system by EDA method

VG1	1.10		VG 11	0.90
VG 2	0.72		VG 12	1.00
VG 3	1.01		VG 13	1.00
VG 4	1.01		VG 14	0.90
VG 5	1.10		VG 15	1.00
VG 6	1.10		VG 16	1.00
VG 7	1.10		VG 17	0.90
VG 8	1.01		VG 18	1.00
VG 9	1.10		VG 19	1.10
VG 10	1.01		VG 20	1.10

T1	1.00		T21	0.90		T41	0.90
T2	1.00		T22	0.90		T42	0.90
T3	1.00		T23	0.90		T43	0.91
T4	1.10		T24	0.90		T44	0.91
T5	1.00		T25	0.90		T45	0.91
T6	1.00		T26	1.00		T46	0.90
T7	1.00		T27	0.90		T47	0.91
T8	1.01		T28	0.90		T48	1.00
T9	1.00		T29	1.01		T49	0.90
T10	1.00		T30	0.90		T50	0.90
T11	0.90		T31	0.90		T51	0.90
T12	1.00		T32	0.90		T52	0.90
T13	1.01		T33	1.01		T53	1.00
T14	1.01		T34	0.90		T54	0.90
T15	1.01		T35	0.90		T55	0.90
T19	1.02		T39	0.90			
T20	1.01		T40	0.90			

Table 4: Optimum real power loss values obtained for practical 191 utility (Indian) system by EDA method.

Real power Loss (MW)	EDA
Min	144.074
Max	147.142
Average	145.008

6. Conclusion

In this paper a novel approach Enhanced Great Deluge Algorithm (EDA) is successfully solved the optimal reactive power problem. In this proposed Enhanced Great Deluge Algorithm (EDA) model, global and local characters of the algorithms are used in a competent way. In order to evaluate validity of the proposed Enhanced Great Deluge Algorithm (EDA) algorithm, it has been tested on standard IEEE 118 & practical 191 bus test systems and compared to other standard reported algorithms. Results show that Enhanced Great Deluge Algorithm (EDA) reduces the real power loss and voltage profiles are within the limits.

References

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

[2] Lee K Y ,Paru Y M , Oritz J L –A united approach to optimal real and reactive power dispatch , IEEE Transactions on power Apparatus and systems 1985: PAS-104 : 1147-1153

[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 :PWRS-2, No. 1, pp.175-182.,1987.

[4] Deeb N, Shahidehpur S.M, Linear reactive power optimization in a large power network using the decomposition approach. IEEE Transactions on power system 1990: 5(2) : 428-435

[5] E. Hobson ,’Network consrained reactive power control using linear programming, ‘ IEEE Transactions on power systems PAS -99 (4) ,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”, IEE Proc; 131C,(3), pp.85-93.

[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] C.A. Canizares, A.C.Z.de Souza and V.H. Quintana, “Comparison of performance indices for detection of proximity to voltage collapse,’’ vol. 11. no.3, pp.1441-1450, Aug 1996.

[9] K.Anburaja, “Optimal power flow using refined genetic algorithm”, Electr.Power Compon.Syst, Vol. 30, 1055-1063, 2002.

[10] D. Devaraj, and B. Yeganarayana, “Genetic algorithm based optimal power flow for security enhancement”, IEE proc-Generation.Transmission and. Distribution; 152, 6 November 2005.

[11] A. Berizzi, C. Bovo, M. Merlo, and M. Delfanti, “A ga approach to compare orpf objective functions including secondary voltage regulation,” Electric Power Systems Research, vol. 84, no. 1, pp. 187 – 194, 2012.

[12] C.-F. Yang, G. G. Lai, C.-H. Lee, C.-T. Su, and G. W. Chang, “Optimal setting of reactive compensation devices with an improved voltage stability index for voltage stability enhancement,” International Journal of Electrical Power and Energy Systems, vol. 37, no. 1, pp. 50 – 57, 2012.

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.886353

[13] P. Roy, S. Ghoshal, and S. Thakur, “Optimal var control for improvements in voltage profiles and for real power loss minimization using biogeography based optimization,” International Journal of Electrical Power and Energy Systems, vol. 43, no. 1, pp. 830 – 838, 2012.

[14] B. Venkatesh, G. Sadasivam, and M. Khan, “A new optimal reactive power scheduling method for loss minimization and voltage stability margin maximization using successive multi-objective fuzzy lp technique,” IEEE Transactions on Power Systems, vol. 15, no. 2, pp. 844 – 851, may 2000.

[15] W. Yan, S. Lu, and D. Yu, “A novel optimal reactive power dispatch method based on an improved hybrid evolutionary programming technique,” IEEE Transactions on Power Systems, vol. 19, no. 2, pp. 913 – 918, may 2004.

[16] W. Yan, F. Liu, C. Chung, and K. Wong, “A hybrid genetic algorithminterior point method for optimal reactive power flow,” IEEE Transactions on Power Systems, vol. 21, no. 3, pp. 1163 – 1169, aug. 2006.

[17] J. Yu, W. Yan, W. Li, C. Chung, and K. Wong, “An unfixed piecewiseoptimal reactive power-flow model and its algorithm for ac-dc systems,” IEEE Transactions on Power Systems, vol. 23, no. 1, pp. 170 –176, feb. 2008.

[18] F. Capitanescu, “Assessing reactive power reserves with respect to operating constraints and voltage stability,” IEEE Transactions on Power Systems, vol. 26, no. 4, pp. 2224–2234, nov. 2011.

[19] Z. Hu, X. Wang, and G. Taylor, “Stochastic optimal reactive power dispatch: Formulation and solution method,” International Journal of Electrical Power and Energy Systems, vol. 32, no. 6, pp. 615 – 621, 2010.

[20] A. Kargarian, M. Raoofat, and M. Mohammadi, “Probabilistic reactive power procurement in hybrid electricity markets with uncertain loads,” Electric Power Systems Research, vol. 82, no. 1, pp. 68 – 80, 2012.

[21] 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.

[22] 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.

[23] IEEE, “The IEEE 30-bus test system and the IEEE 118-test system”, (1993), http://www.ee.washington.edu/trsearch/pstca/.

[24] Jiangtao Cao, Fuli Wang and Ping Li, “An Improved Biogeography-based Optimization Algorithm for Optimal Reactive Power Flow”, International Journal of Control and Automation Vol.7, No.3 (2014), pp.161-176.

*Corresponding author.
E-mail address: gklenin@ gmail.com