Research Journal of Applied Sciences, Engineering and Technology 8(22): 2234-2239, 2014
DOI:10.19026/rjaset.8.1223
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: July 28, 2014 Accepted: September 13, 2014 Published: December 15, 2014

Research Article

Application of Hybrid PSOGSA to Reactive Power Optimization Problem

[1]R. Ambika and [2]R. Rajeswari
[1]Department of EEE, Sri Krishna College of Technology,
[2]Department of EEE, Government College of Technology, Coimbatore, India

Abstract: With the increasing power demand, voltage fluctuations are to be controlled for a reliable and stable power system. On the same way voltage fluctuations create reactive power mismatch in the system. To overcome these conditions we have to perform reactive power optimisation that would balance the reactive power flow of the system. There are several methods and algorithms that serve best for this problem. Among which minimising the real power losses and voltage deviation yields balanced reactive power and for this purpose the most efficient soft computing techniques are used. This study deals with a new approach of hybridisation of two algorithms Particle Swarm Optimisation (PSO) and Gravitational Search Algorithm (GSA). The results are produced on standard IEEE30 bus system for the ORPD problem and prove the best from other algorithms.

Keywords: Gravitational Search Algorithm (GSA), loss minimization, Optimal Reactive Power Dispatch (ORPD), Particle Swarm Optimisation (PSO), voltage deviation minimization

INTRODUCTION

Today's fast moving world is more dependent on electric power. Upcoming technologies are to be inherited in it to bring out evolution in the power system hierarchy. This will increase the reliability and efficiency of the system. On the other hand some blackouts occur in the system due to voltage instability. Voltage instability also causes reactive power mismatch in the system. Reactive power optimisation and voltage stability plays a major role in increasing the efficiency of the system (Majumder, 2013). Reactive power optimisation can be done by real power loss minimisation (Suresh *et al*., 2013) and voltage stability can be achieved by minimising the voltage deviations (Deshmukh *et al*., 2012).

In order to perform the above objectives we have to form an ORPD problem. ORPD problems increase the economy and security of the system (Rabiee and Parniani, 2013). The ORPD problem would minimise the real power losses and voltage deviations subjected to several constraints. The bus voltages, tap setting of transformers, shunt capacitors are the control variables used.

Several soft computing and optimisation theories have evolved that would perform the ORPD problem. Many researches have been done on Artificial Neural Networks (Biserica *et al*., 2012), Genetic Algorithms, Fuzzy Logic Models, Particle Swarm Techniques and Gravitational search algorithms. It is reported in those that evolutionary or heuristic algorithms are more efficient than classical algorithms for solving the ORPD problem (Rabiee and Parniani, 2013; Kennedy and Eberhart, 1995).

In this study, a combination of two algorithms that suits best for the given ORPD problem is introduced i.e. PSO and GSA algorithms are combined. A new hybrid low-level co evolutionary heterogeneous PSOGSA algorithm is applied to the problem and the results are obtained for a standard IEEE30 bus system.

FORMULATION OF THE ORPD PROBLEM

The main objective of this ORPD problem is to optimize the reactive power flow in a power system, by minimizing the real power losses and load bus voltage deviations. It considers a number of constraints such as real and reactive power flow, generator bus voltages, load bus voltages, shunt capacitances, reactive power generation, transformer tap settings and transmission line flow. A multi objective function is formed by combining both the objectives together.

Objective function: The main objective function of this problem is to find the optimal settings of reactive power control variables which includes the generator bus voltages, transformer tap settings and shunt capacitances which minimizes the real power loss and voltage deviation. Hence, the objective function is expressed as in Eq. (1):

$$f = \min \{ wP_L + (1-w)VD \} \quad (1)$$

Corresponding Author: R. Ambika, Department of EEE, Sri Krishna College of Technology, Coimbatore, India

where, w is the weighing factor for real power loss and voltage deviation which is set to 0.7.

- **Real Power Loss minimization (P_L):** The total real power losses of the system is given in Eq. (2):

$$P_L = \sum_{k=1}^{N_l} G_k \left(V_i^2 + V_j^2 - 2V_iV_j\cos(\delta_i - \delta_j)\right) \quad (2)$$

where,
N_l = The total number of transmission lines in the system
G_k = The conductance of the line k
V_i and V_j = The magnitudes of the sending end and receiving end voltages of the line
δi and δj = Angles of the end voltages

- **Load bus Voltage Deviation minimization (VD):** Bus voltage magnitudes are maintained within the allowable limit for quality of service. As shown in Eq. (3) voltage profile is improved by minimizing the deviation of the load bus voltage from the reference value (1.0 p.u.):

$$VD = \sum_{k=1}^{N_{pq}} \left|\left(V_k - V_{ref}\right)\right| \quad (3)$$

Constraints: The following equality and inequality constraints are to be considered in this minimization problem:

- **Equality constraints:**
Load flow constraints: The real and reactive power constraints are according to Eq. (4) and (5) respectively as given below:

$$P_{Gi} - P_{Di} - \sum_{j=1}^{N_B} V_iV_{ij}Y_{ij}\cos(\delta_{ij} + \gamma_j - \gamma_i) = 0 \quad (4)$$

$$Q_{Gi} - Q_{Di} - \sum_{j=1}^{N_B} V_iV_{ij}Y_{ij}\sin(\delta_{ij} + \gamma_j - \gamma_i) = 0 \quad (5)$$

- **Inequality constraints:** Generator bus voltage (V_{Gi}) inequality constraint:

$$V_{Gi}^{min} \leq V_{Gi} \leq V_{Gi}^{max}, i\varepsilon ng$$

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

$$V_{Li}^{min} \leq V_{Li} \leq V_{Li}^{max}, i\varepsilon nl$$

Switchable reactive power compensation (Q_{Ci}) inequality constraint:

$$Q_{Ci}^{min} \leq Q_{Ci} \leq Q_{Ci}^{max}, i\varepsilon nc$$

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

$$Q_{Gi}^{min} \leq Q_{Gi} \leq Q_{Gi}^{max}, i\varepsilon ng$$

Transformer tap setting (T_i) inequality constraint:

$$T_i^{min} \leq T_i \leq T_i^{max}, i\varepsilon nt$$

where, nc, ng and nt are the numbers of the switchable shunt capacitors, generators and transformers taps.

THE STANDARD PSO AND STANDARD GSA

The standard PSO and GSA algorithms are discussed below:

Standard particle swarm optimisation (Yoshida *et al*., 2000): PSO has been developed through simulation of simplified social models. The features of the method are as follows:

- The method is based on researches about swarms such as fish schooling and a flock of birds.
- It is based on a simple concept. Therefore, the computation time is short and it requires few memories.
- It was originally developed for nonlinear optimization problems with continuous variables.

According to the PSO algorithm, birds find food by flocking (not by each individual). This observation leads the assumption that every information is shared inside flocking. Moreover, according to observation of behavior of human groups, behavior of each individual (agent) is also based on behavior patterns authorized by the groups such as customs and other behavior patterns according to the experiences by each individual. This assumption is a basic concept of PSO. PSO is basically developed through simulation of a flock of birds in two-dimension space. The position of each agent is represented by XY-axis position and the velocity (displacement vector) is expressed by vx (the velocity of X-axis) and vy (the velocity of Y-axis). Modification of the agent position is realized by using the position and the velocity information.

The modified velocity of each agent can be calculated using the current velocity and the distance from pbest and gbest as shown below:

$$v_i^{k+1} = w_iv_i^k + c_1rand \times \left(pbest_i - s_i^k\right) + c_2rand \times \left(gbest_i - s_i^k\right) \quad (6)$$

where,
v_i^k = The velocity of agent i at k^{th} iteration
v_i^{k+1} = Modified velocity of agent i
rand = A random number between 0 and 1
s_i^k = The current position of agent i at k^{th} iteration
$pbest_i$ = The *pbest* of agent i
gbest = The *gbest* of the group
w_i = The weight function for velocity of agent
c_i = The weight coefficients for each term

And the current position can be calculated from the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (7)$$

Particle swarm optimisation is extremely simple and effective for wide range of functions (Kennedy and Eberhart, 1995). Conceptually, it seems to lie somewhere between genetic algorithms and evolutionary programming. It is highly dependent on stochastic processes, like evolutionary programming. The adjustment toward *pbest* and *gbest* by the particle swarm optimizer is conceptually similar to the crossover operation utilized by genetic algorithms.

Standard gravitational search algorithm: GSA is a novel heuristic optimization method which has been proposed by Esmat *et al*. (2009). The basic physical theory which GSA is inspired from is the Newton's theory that states: Every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them.

The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion (Esmat *et al*., 2009). More precisely, masses obey the following laws:

Law of gravity: Each particle attracts every other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance between them, R. We use here R instead of R^2, because according to our experiment results, R provides better results than R^2 in all experimental cases.

Law of motion: The current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

The algorithm is summarized as: Initialization of the position of N number of agents is randomly selected and initialized:

$$X_i = \left(x_i^1 \dots x_i^d \dots x_i^n\right) \text{ for } i = 1,2,3 \dots N \quad (8)$$

where, X^d_i represents the position of i^{th} agent in the d^{th} dimension.

Compute best and worst values for each agent at each iteration get fitness value:

$$best(t) = \min_{j\in\{1,\dots m\}} fit_j(t) \quad (9)$$

$$worst(t) = \max_{j=\{1,\dots m\}} fit_j(t) \quad (10)$$

The gravitational constant G at time *t* is:

$$G(t) = G_0 e^{\alpha t/T} \quad (11)$$

where,

G_0 = Set to 1, α to 20

T = The total number of iterations

The gravitational and inertial masses are calculated by the following:

$$M_{ai} = M_{pi} = M_{ii} = M_i, i = 1,2 \dots N$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (12)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \quad (13)$$

where,

M_{ai} = The active gravitational mass of i^{th} agent,

M_{pi} = The passive gravitational mass of the i^{th} agent

M_{ii} = The inertia mass of the i_{th} agent

The total force acting on the i^{th} agent *(F_i^d (t))* is calculated from:

$$F_i^d(t) = \sum_{j\in k_{best}, j\neq i}^{N} rand_j F_{ij}^d(t) \quad (14)$$

where,

k_{best} = The set of k agents with best fitness and becomes 2% of the initial population

F_{ij}^d (t) = The force on agent 'i':

$$F_{ij}^d(t) = G(t)\frac{M_i(t)\times M_j(t)}{R_{ij}(t)+\varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \quad (15)$$

where,

F_{ij}^d (t) = The force on agent 'i' from agent 'j' at d^{th} dimension and t^{th} iteration

R_{ij} (t) = The Euclidian distance between 2 agents 'i' and 'j' at iteration t

G(t) = The calculated gravitational constant for the same iteration

ε = A small constant

Acceralation of the i^{th} agent is:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (16)$$

The velocity and position of the agents for next (t+1) iteration is calculated using the following equation:

$$V_i^d(t+1) = rand_i \times V_i^d(t) + a_i^d(t) \quad (17)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (18)$$

The best fitness value computed at the final iteration is the global fitness of the problem and the position of the corresponding agent at same iteration is the global solution of the agent.

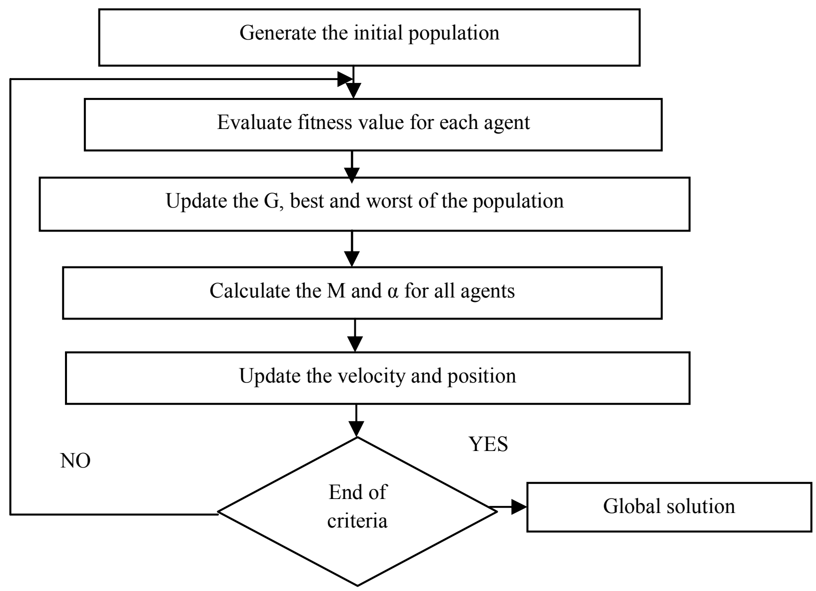

Fig. 1: Flow chart for GSA algorithm

The following flow chart describes the GSA algorithm (Fig. 1).

HYBRID PSOGSA ALGORITHM

Hybridization of different algorithms aims to combine different properties and improve the solution quality. Among the well-known algorithms, PSO and GSA algorithms are the two new algorithms that are used in many fields by researchers and these algorithms are proven to be very powerful optimization tools. Each algorithm has different strong features. PSO generally avoids the solution from trapping into local minima by using its diversity and it's very simple. GSA provides stable convergence characteristics.

Two algorithms can be hybridized in high-level or low-level with relay or co evolutionary method as homogeneous or heterogeneous. In this study, we hybridize PSO with GSA using low-level co evolutionary heterogeneous hybrid. The hybrid is low-level because we combine the functionality of both algorithms. It is co-evolutionary because we do not use both algorithm one after another. In other words, they run in parallel. It is heterogeneous because there are two different algorithms that are involved to produce final results (Mirjalili and Hashim, 2010).

The main objective is to combine the social thinking ability of PSO with the local search capability of GSA. Hence we arrive at a new formula for hybrid PSOGSA for velocity updation as:

$$v_i(t+1) = w \times v_i(t) + c_1^{'} \times rand \times ac_i(t) + c_2^{'} \times rand \times \left(gbest - X_i(t)\right) \quad (19)$$

where,

$v_i(t)$ = The velocity of agent *i* at iteration *t*
$c_j^{'}$ = A weighting factor
w = A weighting function,
rand = A random number between 0 and 1
$ac_i(t)$ = The acceleration of agent *i* at iteration *t*
gbest = The best solution so far

Position updation is done by the following formula:

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (20)$$

First the agents are initialized randomly where each agent is considered as a candidate solution. Then gravitational Mass, gravitational constant, force on each agent are calculated. Next acceleration of the particle is calculated and best solution so far is updated for each iteration. Velocities of all agents are calculated and best positions are identified. When iteration reaches the end criterion the velocity and position updation is stopped. Thus the best solution is obtained.

Figure 2 flow chart describes the process of PSOGSA

RESULTS AND DISCUSSION

The hybrid algorithm is tested on a standard IEEE 30 bus system (Lee *et al*., 1985) using MATLAB. AC load flow is run by MATPOWER simulation software package. MATPOWER (Zimmerman and Gan, 1997) is an open-source power system simulation package for MATLAB. It is used widely in research and education for AC and DC power flow and optimal power flow (OPF) simulations. MATPOWER is designed to give the best performance possible while keeping the code simple to understand and easy to modify (Kristiansen, 2003). And the results are proposed for IEEE 30 bus system. The system has 6 generating buses at 1, 2, 5, 8, 11 and 13. The transformer tap settings are made at 4 lines and shunt capacitors are added at 9 buses.

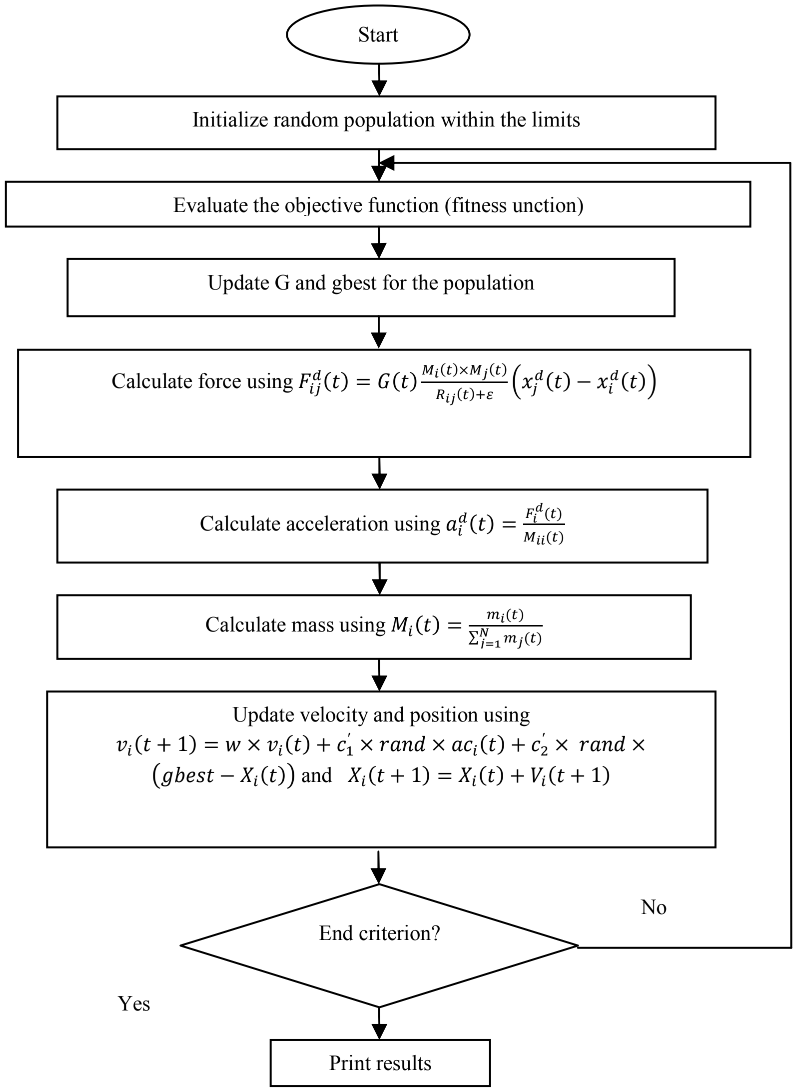

Fig. 2: Flow chart for hybrid PSOGSA algorithm

Table 1: Initial and optimal values of control variables

S.NO	Control variables	Initial value	Optimal value
1	V_{G1}	1.050	1.1000
2	V_{G2}	1.040	1.0903
3	V_{G5}	1.010	1.0637
4	V_{G8}	1.010	1.0699
5	V_{G11}	1.050	0.9872
6	V_{G13}	1.050	0.9991
7	T_{6-9}	1.078	1.1000
8	T_{6-10}	1.069	1.0909
9	T_{4-12}	1.032	1.1000
10	T_{27-28}	1.068	1.0023
11	Q_{10}	0.000	6.5515
12	Q_{12}	0.000	6.4110
13	Q_{15}	0.000	6.4322
14	Q_{17}	0.000	6.4904
15	Q_{20}	0.000	6.4690
16	Q_{21}	0.000	6.4263
17	Q_{23}	0.000	6.4791
18	Q_{24}	0.000	6.5015
19	Q_{29}	0.000	6.4283

Table 2: Simulation results

Values	Power losses (MW)	Voltage deviation
Initial	5.8120	0.981900
Optimal	4.8379	$8.458*10^{-4}$

The limits for the generator voltages are (0.9-1.1) p.u, tap settings are (0.9-1.1) p.u and shunt capacitors are (0-10) MVARs. The constant values are set as follows c_1 = 0.5, c_2 = 1.5, G_0 = 1, α = 20.The test is performed with 50 agents and maximum number of iterations is set to 500. The results are listed in Table 1. The initial and optimal best values for each control variables are listed in the Table 1.

The initial power loss and voltage deviation are 5.812 MW and 0.9819 respectively. And the global best solutions obtained from HYPSOGSA are 4.8379 MW power loss and 8.458e-04 voltage deviation as shown in Table 2. The objective function value is reduced to 3.3868 and results are obtained in 335 sec.

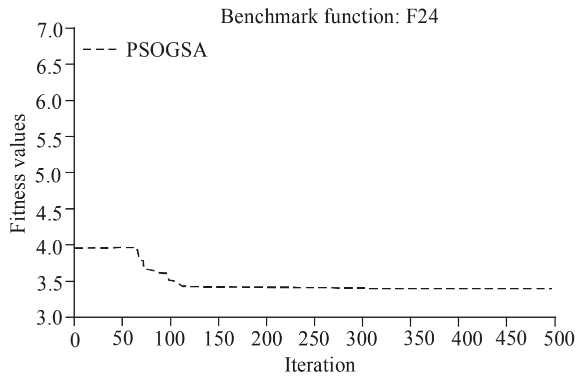

Fig. 3: Graph showing global best solution

Figure 3 illustrates the convergence of global best solution of the algorithm under the given test condition in the Fig. 2.

CONCLUSION

In this study, a new hybrid algorithm is formed combining the strengths of PSO and GSA. The main idea is to integrate the abilities of PSO in exploitation and GSA in exploration. The results show that PSOGSA outperforms both in most function minimization. The results are also proved that the convergence speed of PSOGSA is faster that PSO and GSA.

REFERENCES

Biserica, M., Y. Besanger, R. Caire, O. Chilard and P. Deschamps, 2012. Neural networks to improve distribution state estimation-volt var control performances. IEEE T. Smart Grid, 3(3): 1137-1144.

Deshmukh, S., B. Natarajan and A. Pahwa, 2012. Voltage/VAR control in distribution networks via reactive power injection through distributed generators. IEEE T. Smart Grid, 3(3): 1226-1234.

Esmat, R., N.P. Hossein and S. Saeid, 2009. GSA: A gravitational search algorithm. Inform. Sciences, 179: 2232-2248.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proceeding of IEEE the International Conference on Neural Networks (ICNN). Perth, WA, 4: 1942-1948.

Kristiansen, T., 2003. Utilizing MATPOWER in optimal power flow. Model. Ident. Control., 24(1): 49-59.

Lee, K.Y., Y.M. Park and J.L. Ortiz, 1985. A united approach to optimal real and reactive power dispatch. IEEE T. Power Ap. Syst., PAS-104(5): 1147-1153.

Majumder, R., 2013. Reactive power compensation in single-phase operation of microgrid. IEEE T. Ind. Electron., 60(4): 1403-1416.

Mirjalili, S. and S.Z.M. Hashim, 2010. A new hybrid PSOGSA Algorithm for function optimization. Proceeding of the International Conference on Computer and Information Application (ICCIA, 2010), pp: 374-377.

Rabiee, A. and M. Parniani, 2013. Voltage security constrained multi-period optimal reactive power flow using benders and optimality condition decompositions. IEEE T. Power Syst., 28(2): 696-708.

Suresh, R., C. Kumar, S. Sakthivel and S. Jaisiva, 2013. Application of gravitational search algorithm for real power loss and voltage deviation optimization. Int. J. Eng. Sci. Innov. Technol., 2(1).

Yoshida, H., K. Kawata, Y. Fukuyama, S. Takayama and Y. Nakanishi, 2000. A particle swarm optimization for reactive power and voltage control considering voltage security assessment. IEEE T. Power Syst., 15(4): 1232-1239.

Zimmerman, R.D. and D. Gan, 1997. MATPOWER-a MATLAB Power System Simulation Package. User's Manual, School of Electrical Engineering, Cornell University. Retrieved from: http://www.pserc.cornell.edu/matpower/manual.pdf.