International Journal of Electrical Engineering.
ISSN 0974-2158 Volume 5, Number 2 (2012), pp. 221-229

http://www.irphouse.com

Optimal Reactive Power Dispatch using Bacterial Foraging Algorithm

[1]M. Ettappan M.E., [2]B. Padmanabhan and [3]Dr. M. Rajaram

[1]Head of Department and Associate Professor, Department of Electrical Engineering, Sardar Raja College of Engineering, Tirunelveli Dist, TamilNadu, India
E-mail: ettappanm@rediffmail.com
[2]M.E. Student, Power Systems Engineering,
Sardar Raja College of Engineering, Tirunelveli Dist, TamilNadu India
E-mail: padmanabhan_balu@yahoo.co.in
[3]Vice Chancellor of Anna University of Technology, Tirunelveli, India.

Abstract

Reactive power plays crucial roles in power systems reliability and security. This paper proposes the Bacterial Foraging Algorithm(BFA) applied to optimal reactive power dispatch (ORPD). Optimal reactive power dispatch is a mixed integer, nonlinear optimization problem which includes both continuous and discrete control variables. The proposed algorithm is used to find the settings of control variables such as generator voltages, tap positions of tap changing transformers and the amount of reactive compensation devices to be switched for real power loss minimization and improve the voltage profile in the transmission system. The proposed algorithm is evaluated on an IEEE 30-bus power system, Simulation results show that the proposed approach converges to better solutions much faster than the earlier reported approaches. The optimization strategy is general and can be used to solve other power system optimization problems as well.

Keywords: Reactive Power dispatch; Bacterial Foraging Algorithm; Transmission Open Access

Introduction

Reactive Power Dispatch for improving economy and security of power system operation has received much attention at present. The main objective of optimal reactive power control is to improve the voltage profile and minimizing system real power losses via redistribution of reactive power in the system. In addition, the voltage stability can be enhanced by reallocating reactive power generations. Therefore, the problem of the RPD can be optimized to enhance the voltage stability, improve voltage profile and minimize the system losses as well. The reactive power dispatch problem has a significant influence on secure and economic operation of power systems. Reactive power optimization is a sub problem of the optimal power-flow (OPF) calculation, which determines reactive power outputs of generators, control voltages of PV-buses and tap settings of the under-load tap changing transformers to minimize network power loss. Solving these problems is subject to a number of constraints, such as limits on bus voltages, tap settings of transformers, and number of controllable variables, etc [1].

The reactive power dispatch problem has a significant influence on secure and economic operation of power systems. Reactive power optimization is a sub problem of the optimal power-flow (OPF) calculation[4], which determines all kinds of controllable variables, such as reactive-power outputs of generators and static reactive power compensators, tap ratios of transformers, outputs of shunt capacitors/reactors, etc., and minimizes transmission losses or other appropriate objective functions, while satisfying a given set of physical and operating constraints. Since transformer tap ratios and outputs of shunt capacitors/reactors have a discrete nature, while reactive power outputs of generators and static VAR compensators, bus-voltage magnitudes, and angles are, on the other hand, continuous variables, the reactive power optimization problem can be exactly formulated using a Bacterial Foraging Algorithm, i.e., cast as a nonlinear optimization problem with a mixture of discrete and continuous variables.

Up to now, a number of techniques ranging from classical techniques like gradient-based optimization algorithms to various mathematical programming techniques have been applied to solve this problem [2]-[6]. Recently, due to the basic efficiency of interior-point methods, which offer fast convergence and convenience in handling inequality constraints in comparison with other methods, interior-point linear programming [7], quadratic programming [8], and nonlinear programming [9] methods have been widely used to solve the OPF problem of large-scale power systems. However, these techniques have severe limitations in handling nonlinear, discontinuous functions and constraints, and function having multiple local minima. Unfortunately, the original reactive power problem does have these properties. In all these efforts some or the other simplification has been done to get over the inherent limitations of the solution technique.

Recently, agent-based computation has been studied in the field of distributed artificial intelligence [10] and has been widely used in other branches of computer science [11]. Problem solving is an area that many multiagent-based applications are concerned with. Liu *et al.* [12] introduced an application of distributed techniques for solving constraint satisfaction problem. This paper proposes an efficient BFA Algorithm for solving the reactive power optimization problem.

The rest of this paper is organized as follows: Section II describes the formulation of an Reactive power dispatch problem; while section III explains the standards in BFA. Section IV then details the procedure of handling the BFA. Section V gives the simulation results. Section VI outlines our conclusion.

Problem Formulation

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

$$f_Q = \sum_{k \in N_E} P_{kloss} = \sum_{k \in N_E} g_k (V_i^2 + V_j^2 - 2V_i V_j \cos\theta_{ij}) \tag{1}$$

where $k = (i, j); i \in N_B; j \in N_i$ The symbols of the above equation and in the following context are given in the Nomenclature section. The minimization of the above function is subject to a number of constraints:

$$P_{Gi} - P_{Di} - V_i \sum_{j \in N_i} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) = 0 \quad i \in N_O \tag{2}$$

$$Q_{Gi} - Q_{Di} - V_i \sum_{j \in N_i} V_j (G_{ij} \sin\theta_{ij} + B_{ij} \cos\theta_{ij}) = 0 \quad i \in N_{PQ} \tag{3}$$

and

$$V_i^{\min} \le V_i \le V_i^{\max} \qquad i \in N_B \tag{4}$$

$$T_k^{\min} \le T_k \le T_k^{\max} \qquad k \in N_T \tag{5}$$

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max} \qquad i \in N_G \tag{6}$$

$$Q_{Ci}^{\min} \le Q_{Ci} \le Q_{Ci}^{\max} \qquad i \in N_C \tag{7}$$

$$S_l \le S_l^{\max} \qquad l \in N_l \tag{8}$$

where power flow equations are used as equality constraints, reactive power source installation restrictions, reactive generation restrictions, transformer tap-setting restrictions, bus voltage restrictions and power flow of each branch are used as inequality constraints.

In the most of the nonlinear optimization problems, the constraints are considered by generalizing the objective function using penalty terms. In the reactive power dispatch problem, the generator bus voltages V_{PV} and V_S, the tap position of transformer T, and the amount of the reactive power source installation Q_C are control variables which are self-constrained. Voltages of $PQ\text{–}bus$ V_{PQ} and injected reactive power of $PV\text{–}bus$ Q_G are constrained by adding them as penalty terms to the objective function (1). The above problem is generalized as follows:

$$F_Q = f_Q + \sum_{i \in N_V^{\lim}} \lambda_{Vi} (V_i - V_i^{\lim})^2 + \sum_{i \in N_Q^{\lim}} \lambda_{Gi} (Q_{Gi} - Q_{Gi}^{\lim})^2 \tag{9}$$

Where λ_{Vi} and λ_{Gi} are the penalty factors, and both penalty factors are large positive constants; $V_i^{\lim}$ and $Q_{Gi}^{\lim}$ are defined as,

$$V_i^{\lim} = \begin{cases} V_i^{\max}; & V_i > V_i^{\max} \\ V_i^{\min}; & V_i < V_i^{\min} \end{cases} \tag{10}$$

$$Q_{Gi}^{\lim} = \begin{cases} Q_{Gi}^{\max}; & Q_{Gi} > Q_{Gi}^{\max} \\ Q_{Gi}^{\min}; & Q_{Gi} < Q_{Gi}^{\min} \end{cases} \tag{11}$$

Bacterial Foraging Algorithm

BFA method was invented by Kevin M. Passino motivated by the natural selection which tends to eliminate the animals with poor foraging strategies and favor those having successful foraging strategies [13]. During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. E.coli bacteria's behavior and movement comes from a set of six rigid spinning (100–200 r.p.s) flagella, each driven as a biological motor. An E. coli bacterium alternates through running and tumbling. Running speed is 10–20 µm/s, but they cannot swim straight. Flagella help an *E.coli* bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. After many generations, poor foraging strategies are either eliminated or reshaped into good ones. The foraging strategy is governed basically by four processes namely Chemotaxis, Swarming, Reproduction, Elimination and Dispersal.

Chemotaxis

Chemotaxis process is the characteristics of movement of bacteria in search of food and consists of two processes namely swimming and tumbling. A bacterium is said to be 'swimming' if it moves in a predefined direction, and 'tumbling' if moving in an altogether different direction. Suppose $\theta^i(j,k,l)$ represents i th bacterium at j th chemotactic, k th reproductive and l th elimination dispersal step. $C(i)$ is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may be represented by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{12}$$

where Δ indicates a vector in the random direction whose elements lie in [–1, 1].

Swarming

It is always desired that the bacterium which has searched optimum path of food search should try to attract other bacteria so that they reach the desired that the bacterium which has searched optimum path of food search should try to attract other bacteria so that they reach the desired place more rapidly. Swarming makes the bacteria congregate into groups and hence move a concentric pattern of groups with high bacterial density. Mathematically, swarming can be represented by

$$J_{CC}(\theta,P(j,k,l)) = \sum_{i=1}^{S} J_{CC}(\theta,\theta^i(j,k,l))$$

$$= \sum_{i=1}^{S}[-d_{attract}\exp(-\omega_{attract}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2)] +$$

$$\sum_{i=1}^{S}[h_{repellant}\exp(-\omega_{repellant}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2)] \tag{13}$$

where, J_{CC} is the penalty added to the original cost function. J_{CC} is basically the relative distances of each bacterium from the fittest bacterium. S is the number of bacterium, 'p' represents number of parameters to be optimized, θ_m is the position of the fittest bacterium, $d_{attract}$, $h_{repellant}$, $\omega_{attract}$ and $\omega_{repellant}$ are different coefficients.

Reproduction
In reproduction, population members who have had sufficient nutrients will reproduce and the least healthy bacteria will die. The healthier half of the population replaces with the other half of bacteria which gets eliminated, owing to their poorer foraging abilities. This makes the population of bacteria constant in the evolution process.

Elimination and Dispersal
A sudden unforeseen event may drastically alter the evolution and may cause the elimination and/or dispersion to a new environment. They have the effect of possibly destroying the chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. Elimination and dispersal helps in reducing the behavior of stagnation i.e. being trapped in a premature solution point or local optima.

Algorithm For Bfa Method

Step 1 The following variables are initialized.
Number of bacteria (S) to be used in the search.
Number of parameter (P) to be optimized.
Swimming length N_s.
N_c the number of iteration in a chemotactic loop. ($N_c > N_s$).
N_{re} the no of reproduction.
N_{ed} the no of elimination and dispersal events.
Location of each bacterium P(p, S, *l*) i.e. random numbers on [0-1].
The values of $d_{attract}$, $h_{repellant}$, $\omega_{attract}$ and $\omega_{repellant}$.

Step 2 Elimination-dispersal loop: *l*=*l*+1

Step 3 Reproduction loop: *k*=*k*+1

Step 4 Chemotaxis loop: *j*=*j*+1

Calculate [a] For *i* = 1,2…S take a chemotactic step for bacterium *i* as follows.
[b] Compute the fitness function, *J (i, j, k, l)*.
Let, *J (i, j, k, l)* = *J (i, j, k, l)* + $J_{CC}(\theta^i(j,k,l), P(j,k,l))$ (i.e.

add on the cell – to cell attractant-repellant profile to simulate the swarming behavior)
Where, J_{CC} is defined in (11).

[c] Let $J_{last} = J(i, j, k, l)$ to save this value since we may find a better cost via a run.

Tumble [d] Generate a random vector $\Delta(i) \in \Re^p$ with each element $\Delta_m(i)$, m=1,2,…,p, a random number on [-1, 1].

[e] Move : Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

This results in a step of size C(i) in the direction of the tumble for bacterium i.

[f] Compute $J(i, j+1,k,l)$ and let

$J(i, j+1,k,l) = J(i, j,k,l) + J_{CC}(\theta^i(j+1,k,l), P(j+1,k,l)$.

Swim [g] Let m=0 (counter for swim length).
While m<N_s (if have not climbed down too long).
Let m=m+1.
If $J(i, j+1,k,l) < J_{last}$ (if doing better), let
$J_{last} = J(i, j+1,k,l)$ and let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

And use this $\theta^i(j+1,k,l)$ to compute the new $J(i, j+1,k,l)$ as we did in [f]
Else, let m=N_S. This is the end of the while statement.

[h] Go to next bacterium (i+1) if $i \neq S$ (i.e., go to [b] to process the next bacterium).

Step 5 If j<N_c, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

Step 6 Reproduction [a] For the given k and l, and for each i=1,2,…,S, let

$$J^i_{health} = \sum_{j=1}^{N_C+1} J(i, j,k,l)$$

be the health of the bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters C(i) in order of ascending cost J_{health} (higher cost means lower health).

[b] The S_r bacteria with the highest J_{health} values die and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made are placed at the same

location as their parent).

Step 7 If $k < N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start next generation of the chemotactic loop.

Step 8 Elimination – dispersal: For $i = 1,2,\ldots,S$ with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l<N_{ed}$, then go to step 2; otherwise end.

Simulation Results

To verify the effectiveness and efficiency of the proposed BFA Algorithm based reactive power optimization approach, the IEEE 30-bus power system are used as the test system. The BFA has been implemented in Matlab 7.11.0.584 programming language and numerical tests are carried on a Intel(R) Core(TM) i3 @ 2.53 GHz computer.

IEEE 30-BUS Power System

The IEEE 30-bus system data and operating conditions are given in the [1]. The network consists of 48 branches, six generator-buses, and 22 load-buses. After implementing the BFA to the ORPD problem for different objective functions the results are presented. Table 2 compares optimal transmission loss for the 30-bus IEEE network for different methods after ten runs for each method. The table also shows percentage of power loss decrease with respect to the case that all generator voltages and transformer taps are set to 1 p.u. and reactive compensation devices are set to zero.

Table I: Values of control variables after optimization by HSA, SGA, PSO and BFA

Control Device	HSA	SGA	PSO	BFA
V_1	1.0726	1.0512	1.0313	1.0286
V_2	1.0625	1.0421	1.0114	1.02
V_5	1.0399	1.0322	1.0221	1.0174
V_8	1.0422	0.9815	1.0031	1.0123
V_{11}	1.0318	0.9766	0.9744	0.9532
V_{13}	1.0681	1.1	0.9987	0.9975
T_1	1.01	0.95	0.97	0.9567
T_2	1.00	0.98	1.02	1.053
T_3	0.99	1.04	1.01	1.021
T_4	-0.05	1.02	0.99	0.85
Q_1	0.34	0.12	0.17	0.169
Q_2	0.12	-0.1	0.13	0.103
Q_3	0.10	0.3	0.23	0.265

The results obtained from BFA for power loss reduction are compared with other algorithms such as in [14] HSA, [15] which a DE method is used to solve the optimization problem or CLPSO in [16]. In all of these references some of the constraints and initial settings of the problem are different with the assumed values and constraints.

Table II: Results of Transmission Loss Compared With Results Of [17]

Method	Power Loss (MW)
CGA	25.244
AGA	24.5648
CLPSO	24.5152
L-DE	27.8126
L-SaDE	27.9155
SOA	24.2654
HSA	24.5612
BFA	24.2632

The results presented by the Harmony search algorithm and some other algorithms are better than the result obtained from BFA but it should be regarded that the operations implemented to the data in the search process are very simple. In the future studies modifying the operations used in BFA with other techniques in the form of hybridization for achieving better results than the mentioned algorithms in Table II will be investigated.

Conclusion

The proposed BFA method obtains lesser loss reduction compared to other techniques PSO, SGA, and HSA with less number of population sizes. The minimum loss obtained by BFA is lesser than the minimum loss obtained by other algorithms. The minimum loss value obtained by the proposed BFA is always closer to the average loss value for the test systems. It is observed from the repeated trail runs that BFA converges to near optimal solution with high success rate. The computational results show that the BFA Algorithm can be used for solving the ORPD problems successfully.

References

[1] B.Zhao, C.X.Guo and Y.J.Cao., 2005, "A multiagent-Based particle swarm optimization approach for optimal reactive power dispatch," IEEE Transactions on power systems, 20(2), pp.1070-78.

[2] K. Y. Lee, Y. M. Park, and J. L. Ortiz., 1985, “A united approach to optimal real and reactive power dispatch,” IEEE Trans. Power Syst., 1(2), pp. 1147–1153.

[3] Y. Y. Hong, D. I. Sun, S. Y. Lin, and C. J. Lin., 1990, “Multi-year multi-case optimal AVR planning,” IEEE Trans. Power Syst., 5(4), pp. 1294–1301.

[4] Linda Slimani, Tarek Bouktir., 2006, “An Ant Colony Optimization for solving the optimal power flow problem in medium-scale electrical network”, JEE Trans. Indus elect pow syst., 6(1), pp. 10-15.

[5] N. Deeb and S. M. Shaidepour., 1990, “Linear reactive power optimization in a large power network using the decomposition approach,” IEEE Trans. Power Syst., 5(2), pp. 428–435.

[6] J. R. S. Manlovani and A. V. Garcia., 1995, “A heuristic method for reactive power planning,” IEEE Trans. Power Syst., 11(1), pp. 68–74.

[7] S. Granville., 1994, “Optimal reactive dispatch through interior point methods,” IEEE Trans. Power Syst., 9(1), pp. 136–146.

[8] J. A. Momoh, S. X. Guo, E. C. Ogbuobiri, and R. Adapa., 1994, “The quadratic interior point method solving power system optimization problems,” IEEE Trans. Power Syst., 9(3), pp. 1327–1336.

[9] Y. C. Wu, A. S. Debs, and R. E. Marsten., 1994, “A direct nonlinear predictorcorrector primal-dual interior point algorithm for optimal power flows,” IEEE Trans. Power Syst., 9(2), pp. 876–883.

[10] J. Liu., 2001, “Explorations in Learning Self-Organization, and Adaptive Computation,” Autonomous Agents and Multiagent Systems., Singapore: World Scientific.

[11] J. Liu, Y. Y. Tang, and Y. C. Cao., 1997, “An evolutionary autonomous agents approach to image feature extraction,” IEEE Trans. Evol. Comput., 1(1), pp. 141–158.

[12] J. Liu, H. Jing, and Y. Y. Tang., 2002, “Multiagent oriented constraint satisfaction,” Artificial Intell., 136, pp. 101–144.

[13] Sambarta Dasgupta, Swagatam Das, Ajith Abraham, and Arijit Biswas., 2009, “Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis” IEEE Transactions on Evolutionary Computation, 13(4), pp. 919 to 941.

[14] A.H. Khazali, M. Kalantar., 2011,”Optimal reactive power dispatch based on harmony search algorithm” Electrical Power and Energy Systems, 33, pp. 684–692.

[15] Varadarajan M, Swarup KS., 2008, “Differential evolutionary algorithm for optimal reactive power dispatch,” Electr Power Energy Syst., 30, pp.435–41.

[16] Mahadevan K, Kannan PS., 2010,”Comprehensive learning particle swarm optimization for reactive power dispatch,” Appl Soft Comput., 10, pp.641–52.

[17] Dai Chaohua, Chen Weirong, Zhu Yunfang, Zhang Xuexia., 2009, “Seeker optimization algorithm for optimal reactive power dispatch,” IEEE trans on power syst., 24(3) pp. 1218 - 1231.