INTERNATIONAL JOURNAL OF RESEARCH –
GRANTHAALAYAH
A knowledge Repository

Science

ACTIVE POWER LOSS REDUCTION BY BETTER-QUALITY PARTICLE SWARM OPTIMIZATION ALGORITHM

Dr.K.Lenin [*1]

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

Abstract

In this paper Better-Quality Particle Swarm Optimization (BPSO) algorithm is proposed to solve the optimal reactive power Problem. Proposed algorithm is obtained by combining particle swarm optimization (PSO), Cauchy mutation and an evolutionary selection strategy. The idea is to introduce the Cauchy mutation into PSO in the hope of preventing PSO from trapping into a local optimum through long jumps made by the Cauchy mutation. In order to evaluate the efficiency of the proposed Better-Quality Particle Swarm Optimization (BPSO) algorithm, it has been tested on IEEE 57 bus system. Simulation Results show's that BPSO is more efficient than other reported algorithms in reducing the real power loss.

Keywords: Particle Swarm Optimization; Cauchy Mutation; Optimal Reactive Power; Transmission Loss.

1. Introduction

Different numerical methods have been implemented to solve this optimal reactive power dispatch problem. These consist of 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. In recent times Global Optimization techniques such as genetic algorithms have been proposed to solve the reactive power flow problem [8.9]. 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 [10]. In this paper Better-Quality Particle Swarm Optimization (BPSO) algorithm is proposed to solve the optimal   reactive   power   Problem. Proposed algorithm is obtained by combining particle swarm optimization (PSO), Cauchy mutation and an evolutionary selection strategy. The idea is to introduce the Cauchy mutation into PSO in the hope of

preventing PSO from trapping into a local optimum through long jumps made by the Cauchy mutation.   In order to evaluate the efficiency of the proposed Better-Quality Particle Swarm Optimization (BPSO) algorithm, it has been tested on IEEE 57 bus system. Simulation Results show's that BPSO is more efficient than other reported algorithms in reducing the real power loss.

2. Objective Function

2.1. Active Power Loss

Main aim of the reactive power problem is to reduce the active power loss in the transmission network, which can be described as:

$$F = PL = \sum_{k \in Nbr} g_k \left(V_i^2 + V_j^2 - 2V_i V_j cos\theta_{ij} \right) \tag{1}$$

Where g_k: is the conductance of branch between nodes i and j, Nbr: is the total number of transmission lines in power systems.

2.2. Voltage Profile Improvement

For minimization of the voltage deviation in PQ buses, the objective function turns into:

$$F = PL + \omega_v \times VD \tag{2}$$

Where ω_v: is a weighting factor of voltage deviation.

VD is the voltage deviation given by:

$$VD = \sum_{i=1}^{Npq} |V_i - 1| \tag{3}$$

2.3. Equality Constraint

The equality constraint of the Reactive power problem is represented by the power balance equation, and can be written as, where the total power generation must cover the total power demand and total power loss:

$$P_G = P_D + P_L \tag{4}$$

Where, P_G - Total Power Generation, P_D-Total Power Demand, P_L – Total Power Loss.

2.4. Inequality Constraints

Inequality constraints define the limitations in power system components and power system security. Upper and lower bounds on the active power of slack bus, and reactive power of generators are written as follows:

$$P_{gslack}^{min} \leq P_{gslack} \leq P_{gslack}^{max} \tag{5}$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max} , i \in N_g \tag{6}$$

Upper and lower bounds on the bus voltage magnitudes are described as follows:

$$V_i^{min} \leq V_i \leq V_i^{max} , i \in N \tag{7}$$

Upper and lower bounds on the transformers tap ratios are given as follows:

$$T_i^{min} \leq T_i \leq T_i^{max} , i \in N_T \tag{8}$$

Upper and lower bounds on the compensators reactive powers are written as follows:

$$Q_c^{min} \leq Q_c \leq Q_C^{max} , i \in N_C \tag{9}$$

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.

3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) [11, 12] is motivated from the social behaviour of organisms, such as bird flocking and fish schooling. Particles "fly" through the search space by following the previous best positions of their neighbours and their own previous best positions. Each particle is represented by a position and a velocity which are updated as follows:

$$X_{id}' = X_{id} + V_{id}' \tag{10}$$

$$V_{id}' = \omega V_{id} + \eta_1 rand(P_{id} - X_{id})(P_{id} - X_{id}) + \eta_2 rand()\left(P_{gd} - X_{id}\right) \tag{11}$$

Where X_{id}' and X_{id} represent the current and the previous positions of *id*th particle, V_{id} and V_{id}' are the previous and the current velocity of *id*th particle, P_{id} and P_{gd} are the individual's best position and the best position found in the whole swarm so far respectively. $0 \leq \omega < 1$ is an inertia weight which determines how much the previous velocity is preserved, η_1 and η_2 are acceleration constants, *rand()* generates random number from interval [0,1]. In PSO, each particle shares the information with its neighbours. The updating equations (10) and (11) show that PSO combines the cognition component of each particle with the social component of all the particles in a group.

The social component suggests that individuals ignore their own experience and adjust their behaviour according to the previous best particle in the neighbourhood of the group. On the other hand, the cognition component treats individuals as isolated beings and adjusts their behaviour only according to their own experience. Although the speed of convergence is very fast, many experiments have shown that once PSO traps into local optimum, it is difficult for PSO to jump out of the local optimum. Ratnaweera et.al. [13] State that lack of population diversity in PSO

algorithms is understood to be a factor in their convergence on local optima. Therefore, the addition of a mutation operator to PSO should enhance its global search capacity and thus improve its performance. A first attempt to model particle swarms using the quantum model (QPSO) was carried out by Sun et.al, [14]. In a quantum model, particles are described by a wave function instead of the standard position and velocity. The quantum Delta potential well model and quantum harmonic oscillators are commonly used in particle physics to describe the stochastic nature of particles. In their studies [15], the variable of *gbest* (the global best particle) and *mbest* (the mean value of all particles' previous best position) is mutated with Cauchy distribution respectively, and the results show that QPSO with *gbest* and *mbest* mutation both performs better than PSO. The work of R. A. Krohling et.al,[16, 17] showed that how Gaussian and Cauchy probability distribution can improve the performance of the standard PSO. Recently, evolutionary programming with exponential mutation has also been proposed [18]. In order to prevent PSO from falling in a local optimum, a prompt PSO (PPSO) is proposed by introducing a Cauchy mutation operator in this paper. Because the expectation of Cauchy distribution does not exist, the variance of Cauchy distribution is infinite. Some researches [19, 20] have indicated that the Cauchy mutation operator is good at the global search for its long jump ability.

4. Better-Quality Particle Swarm Optimization (BPSO) Algorithm by Cauchy Mutation and Natural Selection Strategy

4.1. Cauchy Mutation

From the mathematic theoretical analysis of the trajectory of a PSO particle [21-23], the trajectory of a particle X_{id} converges to a weighted mean of P_{id} and P_{gd}. Whenever the particle converges, it will "fly" to the personal best position and the global best particle's position. According to the update equation, the personal best position of the particle will gradually move closer to the global best position. Therefore, all the particles will converge onto the global best particle's position. This information sharing mechanism makes PSO have a very fast speed of convergence. Meanwhile, because of this mechanism, PSO can't guarantee to find the global minimal value of a function. In fact, the particles usually converge to local optima. Without loss of generality, only function minimization is discussed here. Once the particles trap into a local optimum, in which P_{id} can be assumed to be the same as P_{gd}, all the particles converge on P_{gd}. At this condition, the velocity update equation becomes:

$$V'_{id} = \omega V_{id} \tag{12}$$

When the iteration in the equation (12) goes to infinite, the velocity of the particle V_{id} will be close to *0* because of $0 \leq \omega < 1$. After that, the position of the particle X_{id} will not change, so that PSO has no capability of jumping out of the local optimum. It is the reason that PSO often fails on finding the global minimal value. To overcome the weakness of PSO discussed at the beginning of this section, the Cauchy mutation is incorporated into PSO algorithm. The basic idea is that, the velocity and positions of a particle are updated not only according to (10) and (11), but also according to Cauchy mutation as follows:

$$V'_{id} = V_{id} exp(\delta) \tag{13}$$

$$X'_{id} = X_{id} + V'_{id}\delta_{id} \tag{14}$$

Where δ and δ_{id} denote Cauchy random numbers since the expectation of Cauchy distribution doesn't exist, the variance of Cauchy distribution is infinite so that Cauchy mutation could make a particle have a long jump. By adding the update equations of (13) and (14), Better-Quality Particle Swarm Optimization (BPSO) algorithm greatly increases the probability of escaping from the local optimum. In standard PSO, the position of a particle is updated according to equations (10) and (11). That is, for each particle，there is nowhere to move but following the direction of the best particle, and the flying direction is nearly determinate through the generation. From the above analysis of PSO, the particles incline to converge on a local optimum.

4.2. Natural Selection Strategy

In the standard PSO, all particles are directly updated by their offspring no matter whether they are improved. If a particle moves to a better position, it can be replaced by the updated. However if it moves to a worse position, it is still replaced by its offspring. In fact, the most particles fly to worse positions for most cases; therefore the whole swarm will converge on local optima. Like evolutionary algorithms, FPSO introduces an evolutionary selection strategy in which each particle survives according to a natural selection rule. Therefore, the particle's position at the next step is not only due to the position update but also the evolutionary selection. Such strategy could greatly reduce the probability of trapping into local optimum. The evolutionary selection strategy is carried out as follows. Assume the size of the swarm is m, pair-wise comparison over the union of parents and offspring ($1,2,…2m$) is made. For each particle, q opponents are randomly chosen from all parents and offspring with equal probability. If the fitness of particle i is less than its opponent, it will receive a "win". Then select m particles that have the more winnings to be the next generation.

The detail of the selection framework is as follows:
Step1: For each particle of parent and offspring, assign *win[i]=0.*
Step2: Randomly select q particles (opponents) for each particle in parent and offspring.
Step3: For each particle, compare it with its q opponents. For particle *i,* if the fitness of its opponent j is larger than particle i , then *win[i]++.*
Step4: Select m particles that have the more winnings to be the next generation.

BPSO Algorithm for solving optimal reactive power problem
Step1: Generate the initial particles by randomly generating the position and velocity for each particle.
Step2: Evaluate each particle's fitness.
Step3: For each particle, if its fitness is smaller than its previous best(P_{id}) fitness, update P_{id} .
Step4: For each particle, if its fitness is smaller than the best one (P_{gd}) of all the particles, update *Pgd*.
Step5: For each particle, do
   1) Generate a new particle t according to the formula (10) and (11).
   2) Generate a new particle *t'* according to the formula (13) and (14).
   3) Compare t with *t'* chose the one with smaller fitness to be the offspring.

Step6: Generate the next generation according to the above evolutionary selection strategy.
Step7: if the stop criterion is satisfied, then stop, else go to Step 3.

5. Simulation Results

At first Better-Quality Particle Swarm Optimization (BPSO) algorithm has been tested in standard IEEE-57 bus power system. The reactive power compensation buses are 18, 25 and 53. Bus 2, 3, 6, 8, 9 and 12 are PV buses and bus 1 is selected as slack-bus. The system variable limits are given in Table 1.

The preliminary conditions for the IEEE-57 bus power system are given as follows:

P_{load} = 12.120 p.u. Q_{load} = 3.063 p.u.

The total initial generations and power losses are obtained as follows:

$\sum P_G$ = 12.462 p.u. $\sum Q_G$ = 3.3154 p.u.

P_{loss} = 0.25862 p.u. Q_{loss} = -1.2068 p.u.

Table 2 shows the various system control variables i.e. generator bus voltages, shunt capacitances and transformer tap settings obtained after optimization which are within the acceptable limits. In Table 3, shows the comparison of optimum results obtained from proposed methods with other optimization techniques. These results indicate the robustness of proposed approaches for providing better optimal solution in case of IEEE-57 bus system.

Table 1: Variable Limits

Reactive Power Generation Limits						
Bus no	1	2	3	6	8	9	12
Qgmin	-1.4	-.015	-.02	-0.04	-1.3	-0.03	-0.4
Qgmax	1	0.3	0.4	0.21	1	0.04	1.50

Voltage And Tap Setting Limits						
vgmin	Vgmax	vpqmin	Vpqmax	tkmin	tkmax
0.9	1.0	0.91	1.05	0.9	1.0

Shunt Capacitor Limits		
Bus no	18	25	53
Qcmin	0	0	0
Qcmax	10	5.2	6.1

Table 2: Control variables obtained after optimization

Control Variables	BPSO
V1	1.1
V2	1.034
V3	1.032
V6	1.026
V8	1.020

V9	1.008
V12	1.010
Qc18	0.0662
Qc25	0.201
Qc53	0.0470
T4-18	1.002
T21-20	1.040
T24-25	0.862
T24-26	0.870
T7-29	1.052
T34-32	0.870
T11-41	1.011
T15-45	1.031
T14-46	0.912
T10-51	1.021
T13-49	1.062
T11-43	0.910
T40-56	0.900
T39-57	0.950
T9-55	0.950

Table 3: Comparison results

S.No.	Optimization Algorithm	Finest Solution	Poorest Solution	Normal Solution
1	NLP [24]	0.25902	0.30854	0.27858
2	CGA [24]	0.25244	0.27507	0.26293
3	AGA [24]	0.24564	0.26671	0.25127
4	PSO-w [24]	0.24270	0.26152	0.24725
5	PSO-cf [24]	0.24280	0.26032	0.24698
6	CLPSO [24]	0.24515	0.24780	0.24673
7	SPSO-07 [24]	0.24430	0.25457	0.24752
8	L-DE [24]	0.27812	0.41909	0.33177
9	L-SACP-DE [24]	0.27915	0.36978	0.31032
10	L-SaDE [24]	0.24267	0.24391	0.24311
11	SOA [24]	0.24265	0.24280	0.24270
12	LM [25]	0.2484	0.2922	0.2641
13	MBEP1 [25]	0.2474	0.2848	0.2643
14	MBEP2 [25]	0.2482	0.283	0.2592
15	BES100 [25]	0.2438	0.263	0.2541
16	BES200 [25]	0.3417	0.2486	0.2443
17	Proposed BPSO	0.22042	0.23026	0.22210

6. Conclusion

In this paper, Better-Quality Particle Swarm Optimization (BPSO) algorithm has been solved optimal reactive power problem. The idea is to introduce the Cauchy mutation into PSO in the

hope of preventing PSO from trapping into a local optimum through long jumps made by the Cauchy mutation.   In order to evaluate the efficiency of the proposed Better-Quality Particle Swarm Optimization (BPSO) algorithm, it has been tested on IEEE 57 bus system. Simulation Results show's that BPSO is more efficient than other reported algorithms in reducing the real power loss.

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

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

[10]   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.

[11]   J. Kennedy and R. C. Eberhart, Particle Swarm Optimization, IEEE International Conference on Neural Networks, pp.1942-1948, 1995.

[12]   R. C. Eberhart and J. Kennedy, A New Optimizer Using Particle Swarm Theory, Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp.39-43, 1995.

[13]   A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation, vol. 8,no. 3, pp. 240–255, 2004.

[14]   J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in Jing Liu, Wenbo Xu, Jun Sun, Quantum-behaved particle swarm optimization with mutation operator. Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence Pages: 237 - 240 , 2005

[15]   R. A. Krohling, Gaussian particle swarm with jumps, in Proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, UK,pp. 1226-1231, 2005.

[16]   R. A. Krohling, L. dos Santos Coelho, PSO-E: Particle Swarm with Exponential Distribution , in Proceedings of the IEEE Congress on Evolutionary Computation, pp1428- 1433, July 2006.

[17]   H. Narihisa, T. Taniguchi, M. Ohta, and K. Katayama, Evolutionary Programming with Exponential Mutation, in Proceedings of the IASTED Artificial Intelligence and soft Computing, Benidorn, Spain, pp. 55-50, 2005.

[18]   X. Yao and Y. Liu. Fast evolutionary programming, Proc. of the Fifth Annual Conference on Evolutionary Programming (EP'96), San Diego, CA, USA, 29/2-2/3/96. (1996).451-460, the MIT Press.

[19] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster. IEEE Trans. Evolutionary Computation 82–102, 1999.

[20] M. Clerc and J. Kennedy, the Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space, IEEE Trans. on Evolutionary Computation, Vol.6, pp: 58-73, 2002.

[21] G. Pampara, N. Franken, A. P. Engelbrecht, Combining Particle Swarm Optimisation with angle modulation to solve binary problems. In proceedings of the IEEE Congress on Evolutionary Computation, pages 89- 96, Sept. 2005.

[22] F. van den Bergh. An Analysis of Particle Swarm Optimizers. PhD thesis, Department of Computer Science, University of Pretoria, South Africa, 2002.

[23] Chaohua Dai, Weirong Chen, Yunfang Zhu, and Xuexia Zhang, "Seeker optimization algorithm for optimal reactive power dispatch," IEEE Trans. Power Systems, Vol. 24, No. 3, August 2009, pp. 1218-1231.

[24] J. R. Gomes and 0. R. Saavedra, "Optimal reactive power dispatch using evolutionary computation: Extended algorithms," IEE Proc.-Gener. Transm. Distrib. Vol. 146, No. 6. Nov. 1999.

[25] Proceedings of the IEEE Congress on Evolutionary Computation, Portland, Oregon USA, pp. 325-331, 2004.

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