International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 2 040– 042

Improved Dynamic PSO for Optimal Reactive Power Flow Control

Mr. Altaf Q. H. Badar
Electrical Engineering Department
Anjuman College of Engg and Tech
Nagpur, India
altaf_badar@rediffmail.com

Dr. B. S. Umre and Dr. A. S. Junghare
Electrical and Electronics Engineering Department
Visvesvaraya National Institute of Technology
Nagpur, India

Abstract**—** Active power loss in an electrical system can be reduced by controlling the flow of reactive power in the system. The reactive power flow in a system can be controlled through generator voltages, transformer tap setting and capacitor banks. The problem of reactive power flow control is a non linear multidimensional search space problem. Different optimization techniques have been implemented to obtain an optimal combination of these control variables to minimize the active power loss. Particle swarm optimization is one of the most robust and adaptive technique used for minimizing active power loss. Dynamic PSO is an improved variant of PSO. This paper further suggests addition of inertia weight to Dynamic PSO for faster convergence. The proposed method is implemented on IEEE 6 bus system.

Keywords- *Particle Swarm Optimization; dynamic Particle Swarm Optimization; Reactive power optimization; Active power loss minimization; inertia weight;*

*****

I. INTRODUCTION

Reactive power flow control in an electrical system is of paramount importance. Minimizing the flow of reactive power in the system improves the voltage profile, increases reliability, provides economic operation, reduces active power losses, etc. The reactive power flow in an electrical system can be controlled by generator voltages, transformer tap settings and capacitor banks.

The reactive power flowing in an electrical system contributes to the losses taking place in the system. By controlling the flow of reactive power in the system, the active power losses can be reduced. The parameters through which reactive power flow can be controlled are called as control variables. An optimal combination of these control variables is required to achieve optimal reactive power flow in the system.

Various artificial optimization techniques have been used to solve the reactive power flow control optimization problem like particle swarm optimization (PSO) [1-3], genetic algorithm (GA) [4], ant colony optimization (ACO) [5,6], tabu search (TS) [7], differential evolution (DE) [8-10], artificial bee colony (ABC) [11], etc. These methods are derived from various biological processes. Improved and hybrid versions of these methods have also been introduced through the years to solve multidimensional optimization problem.

The reactive power flow problem is a non linear multidimensional optimization problem. The complexity of the problem increases as the number of control variables increase. The optimization technique should be robust and adaptable to obtain the optimal solution of such a problem.

Particle swarm optimization is a widely studied and researched optimization technique. It is simple in its implementation. Particle Swarm optimization (PSO) was introduced in 1995 by Kennedy and Eberhart [12]. Dynamic PSO [1] is an improved variant of PSO and tries to overcome an inherent demerit of PSO caused due to constants present in the calculation of velocity for particles.

The term of inertia weight was introduced in PSO [13] to control the exploration and exploitation of the search space by the particles. For global search the inertia weight is kept large whereas for local search the inertia weight is kept small. Early studies and methods to calculate inertia weight are presented in [14, 15]. Inertia Weight has been a topic of discussion in various researches and review studies [16-26].

This paper proposes to introduce the inertia weight term in Dynamic PSO to improve its performance and speed of convergence. Dynamic PSO with inertia weight has been implemented on IEEE 6 bus system and compared with original dynamic PSO. Inertia weight has been introduced in dPSO using three different methods.

The reactive power flow control problem has been dealt with in detail in [1].

II. PARTICLE SWARM OPTIMIZATION AND DYNAMIC PSO

Particle Swarm optimization has been variedly used in various fields for optimization. It is a robust and easily adaptable method. PSO emulates the behaviour of birds in a flock or fish in a school. The animals try to optimize their movements to gather food, avoid predators and seek mates. In [12], it is also mentioned that the PSO tries to emulate a human behaviour which is based on social and cognitive models. It is easy to emulate a human behaviour since two persons can have same level of thoughts without being at the same place physically, whereas the same is not true for the birds or fishes.

In PSO, a number of random particles are generated. Each particle represents a specific combination of control variables and is a candidate solution itself. The particles travel through the multidimensional search space to find the optimal solution. The movement of the particles is dependent on its previous velocity, individual best and global best values. Previous velocity represents the direction of the recent movement of a particle in multidimensional search space. Individual best of a particle is a position achieved by that particle during the search process having best objective function value. Global best is a position achieved by any particle within the swarm which has the best recorded objective function value. The effect of these parameters on the velocity of a particle is further controlled by constants like constriction factor (K) and acceleration coefficients (c1 and c2). The values of these factors are specified as K=0.729, c1=c2=2.04 in [17]. These values are applicable to all optimization problems being solved by PSO. Since all optimization problems are not same these factors may or may not be optimal for the problem being considered [18]. Many authors and researchers have tried to find out better options to overcome this demerit of PSO. Dynamic PSO is an improvement in PSO which eliminates the usage of these factors from the calculation of velocity.

The velocity of a 'd' dimension particle in dPSO is calculated as given in equation (1):

$$v_{id} = (f(p_{id}) - f(x_{id})) * (p_{id} - x_{id}) * sf_1 + (f(p_{gd}) - f(x_{id})) * (p_{gd} - x_{id}) * sf_2 + rand() * signis() * sf_3 \quad (1)$$

where v_{id} → velocity of the particle 'i'.
$f(p_{id})$ → objective function value of individual best of particle 'i'.
$f(p_{gd})$ → objective function value of global best.
$f(x_{id})$ → objective function value of the particle 'i'.
p_{id} → individual best position of particle 'i'.
p_{gd} → global best position.
x_{id} → current position of the particle 'i'.
rand() → function used for generating random number between 0 and 1.
signis() → used to randomly generate +1 or -1
sf_1, sf_2, sf_3 → scaling factors for different terms.

In [1], it was observed that the dPSO was able to perform better than PSO [17].

III. Improved dPSO

The new position of particles in dPSO is calculated without using the constriction factor and the acceleration coefficients. It is however felt that the movement of particles can be better coordinated by introducing the inertia weight term in dPSO method. The inertia weight is used to control the velocity of particles and their movement. A higher value of inertia weight is used for global search or exploration whereas a lower value of inertia weight is used for local search or exploitation.

It has been proposed and testified in earlier research works [16, 17 and 19] that the inertia weight should have a higher value at the beginning of the search process and should reduce towards its end. The comparison between the effect of inertia weight and constriction factor on performance of PSO is presented in [17]. A detailed study in different inertia weight values and constriction factor is also presented in [19]. An adaptive inertia weight calculation method is presented in [20]. The inertia weight calculation as increasing and decreasing sigmoid function is implemented in [21, 22]. An in depth review study on different methods inertia weight calculation methods and their effect on performance of PSO is discussed in [23, 24 and 25]. In [26], the various inertia weight calculation methods are discussed and their performances are compared.

The new position of particles in dPSO is calculated using the following equations (2a) and (2b):

$$v_{id} = \omega * v_{id} + (f(p_{id}) - f(x_{id})) * (p_{id} - x_{id}) * sf_1 + (f(p_{gd}) - f(x_{id})) * (p_{gd} - x_{id}) * sf_2 + rand() * signis() * sf_3 \quad (2a)$$

$$x_{id} = x_{id} + v_{id} \quad (2b)$$

where ω → inertia weight

The dPSO method [1], initially did not have any inertia weight and hence the 'previous velocity' of a particle was not required for calculating its current velocity. In the improved method the inertia weight term has been added to enhance its performance. The different methods implemented for calculation of inertia weight are enlisted in Table I.

TABLE I. Inertia Weight Calculation Methods

Sr. No.	Method	Explanation
1	ω = 0.729	Inertia weight has a constant value of 0.729
2	ω = 0.9 to 0.4	Inertia weight reduces linearly from 0.9 to 0.4 through the execution of the search process.
3	ω = 0.5 + rnd/2	Inertia weight is a random number between 0.5 - 1, at any point during the search process. The rnd function generates a random number from 0 to 1.

There are other methods of calculating inertia weight which are not considered because of complexity and increase in time required for obtaining the optimal results.

IV. Results

The improved dynamic PSO method was implemented on standard IEEE 6 bus system. The IEEE 6 bus system has 6 control variables: V_1, V_2, T_{43}, T_{56}, Q_4, and Q_6. The data of IEEE 6 bus system is given in Table II. Table III shows the minimum and maximum limits of these control variables.

TABLE II. IEEE 6 Bus System Data

Start Bus	End Bus	Branch Impedance	Transformer Tap
1	6	0.23+j0.518	
1	4	0.080+j0.370	
4	6	0.097+j0.407	
5	2	0.282+j0.640	
2	3	0.723+j1.050	
6	5	0.000+j0.300	1.025
4	3	0.000+j0.133	1.1

TABLE III. Control Variable Constraints

	Transformer Tap Setting	Generator Bus Voltage		Capacitor Bank (MVAR)	
	T_{65}, T_{43}	V_1	V_2	Q_4	Q_6
Minimum Limit	0.910	1.0	1.1	0.0	0.0
Maximum Limit	1.110	1.1	1.15	5.0	5.5
Step Value	0.125	0.01	0.005	0.5	0.5

The initial and optimal values of control variables are given in Table IV.

TABLE IV. Initial and Optimal Values of Control Variables

	V_1	V_2	Q_4	Q_6	T_{43}	T_{65}
Initial State	1.05	1.1	0	0	1.1	1.025
Optimal State	1.1	1.15	5	5.5	0.9475	0.935

The performance of different methods of calculation of inertia weight is compared on their ability to obtain optimal solution in less number of iterations. The average values of iteration number at which the different inertia weight calculation methods are able to reach the optimal solution are presented in Table V.

TABLE V. Average Iteration Number for Obtaining Optimal Result

No of Particles	Without	0.729	Reducing	0.5+rnd/2

ISSN: 2321-8169
040– 042

20	6.6	5.1	5	4.9
25	5.9	4.6	4.6	4.5

From Table V, it can be observed that ω = (0.5 + rnd/2) method of calculating inertia weight in dPSO technique generates the best result. This method is able obtain optimal result in lesser number of iterations as compared to other methods. The above results are presented based on an average of 10 readings for each method. The average number of iterations required to obtain optimal result is reduced by about 15% - 20% on adding the inertia weight term in dPSO.

V. Conclusion

The active power loss minimization of a system is a non linear multidimensional search space problem. Various optimization methods have been implemented to solve the problem. Particle Swarm optimization is one such method and Dynamic PSO is its improved variant. The performance of dPSO has been observed to improve with the introduction of inertia weight term in calculation of particle velocity. Different methods of inertia weight calculation have been implemented and compared.

References

[1] Altaf Q. H. Badar, B. S. Umre and A. S. Junghare, "Reactive power control using dynamic Particle Swarm Optimization for real power loss minimization",International Journal of Electrical Power & Energy Systems, vol.41, Oct 2012, pp. 133-136.

[2] M. A. Abido, "Optimal power flow using Particle Swarm Optimization", International Journal of Electrical Power & Energy Systems, vol. 24, 2002, pp. 563-571.

[3] A. H. Mantawy and M. S. Al-Ghamdi, "A new reactive power optimization algorithm", Power Tech Conference Proceedings, IEEE, vol. 4, 2003, pp. 6-.

[4] Kenji Iba, "Reactive power optimization by Genetic Algorithm" IEEE Transactions on Power Systems, vol. 9, 1994, pp. 685-692.

[5] K. Lenin and M. R. Mohan, "Ant Colony search algorithm for optimal reactive power optimization", Serbian journal of electrical engineering, vol. 3, 2006, pp. 77-88.

[6] Ibrahim Oumarou, Daozhou Jiang, and Cao Yijia, "Optimal reactive power optimization by Ant Colony search algorithm", Fifth International Conference on Natural Computation (ICNC'09), vol. 3, 2009, pp. 50-55.

[7] M. A. Abido, "Optimal power flow using Tabu Search algorithm", Electric Power Components and Systems, Taylor and Francis, vol. 30, 2002, pp. 469-483.

[8] A. A. Ela, M. A. Abido, and S. R. Spea, "Differential evolution algorithm for optimal reactive power dispatch", Electric Power Systems Research, Elsevier, vol. 81, 2011, pp. 458-464.

[9] G. A. Bakare, G. Krost, G. K. Venayagamoorthy and U. O. Aliyu, "Differential evolution approach for reactive power optimization of Nigerian grid system", Power Engineering Society General Meeting, IEEE, 2007, pp. 1-6.

[10] M. Varadarajan and K. S. Swarup, "Network loss minimization with voltage security using Differential Evolution", Electric Power Systems Research, Elsevier, vol. 78, 2008, pp. 815-823.

[11] Ali Ozturk, Serkan Cobanli, Pakize Erdogmus and Salih Tosun, "Reactive power optimization with Artificial Bee Colony algorithm", Scientific Research and Essays, Academic Journals, vol. 5, 2010, pp. 2848-2857.

[12] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization", International Conference on Neural Networks, IEEE , vol.4, Nov 1995, pp. 1942-1948, doi:10.1109/ICNN.1995.488968.

[13] Yuhui Shi and Russell C. Eberhart, "Parameter selection in Particle Swarm Optimization", Evolutionary Programming VII, Springer, 1998, pp. 591-600.

[14] Yuhui Shi and Russell Eberhart, "A modified particle swarm optimizer", IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence., 1998, pp. 69-73.

[15] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization" , Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99), IEEE, vol. 3, 1999, pp. -1957, doi:10.1109/CEC.1999.785513.

[16] Yuhui Shi and R. C. Eberhart, "Empirical study of particle swarm optimization", Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99), IEEE, vol. 3, 1999, pp. -1950, doi:10.1109/CEC.1999.785511.

[17] Russ C. Eberhart, and Yuhui Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", Proceedings of the 2000 Congress on Evolutionary Computation, IEEE, vol. 1, 2000, pp. 84-88.

[18] Anthony Carlisle and Gerry Dozier, "An off-the-shelf PSO", Proceedings of the workshop on particle swarm optimization, Indianapolis, vol. 1, 2001, pp. 1-6.

[19] R.C. Eberhart and Yuhui Shi, "Particle swarm optimization: developments, applications and resources", Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, 2001, pp. 81-86, doi:10.1109/CEC.2001.934374.

[20] Zheng Zhou and Yuhui Shi, "Inertia Weight Adaption in Particle Swarm Optimization Algorithm" Advances in Swarm Intelligence, Springer Berlin Heidelberg, vol. 6728, 2011, pp. 71-79, doi:10.1007/978-3-642-21515-5_9.

[21] Andi Adriansyah and Shamsudin H. M. Amin, "Analytical and empirical study of particle swarm optimization with a sigmoid decreasing inertia weight", Regional Postgraduate Conference on Engineering and Science, School of Postgraduate Studies, UTM, 2006, pp. 247-252.

[22] Reza Firsandaya Malik, Tharek Abdul Rahman, Siti Zaiton Mohd Hashim and Razali Ngah, "New particle swarm optimizer with sigmoid increasing inertia weight", International Journal of Computer Science and Security, vol. 1, 2007, pp. 35-44.

[23] Riccardo Poli, "Analysis of the publications on the applications of particle swarm optimisation", Journal of Artificial Evolution and Applications, Hindawi Publishing Corp., vol. 2008, 2008, pp. 3.

[24] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", IEEE Transactions on Evolutionary Computation, vol. 12, Apr. 2008, pp. 171-195, doi:10.1109/TEVC.2007.896686.

[25] Hu Xiaohui, Yuhui Shi and R. Eberhart, "Recent advances in particle swarm", Congress on Evolutionary Computation (CEC'2004), IEEE, vol. 1, Jun. 2004, pp. 90-97, doi:10.1109/CEC.2004.1330842.

[26] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia Weight strategies in Particle Swarm Optimization", Third World Congress on Nature and Biologically Inspired Computing (NaBIC), Oct. 2011, pp. 633-640, doi=10.1109/NaBIC.2011.6089659.