International Journal of communication and computer Technologies, ISSN: 2278-9723

Available at http://www.ijccts.org

Particle Swarm Optimization for Solving Economic Load Dispatch Problem

Priyanka Sinha
Gyan Vihar School of Engineering and Technology,

Received: 16-03-2013, **Revised:** 27-04-2013, **Accepted:** 26-05-2013, **Published online:** 21-06-2013

ABSTRACT

This work deals with the optimization of power flow through interconnected system. The work deals with efficient and reliable evolutionary based approach to solve the economic load dispatch (ELD) with line flows and voltage constraints. The work employs PSO algorithm for ELD. PSO is a robust, stochastic computational technique based on movement and intelligence of swarm. The work introduces a conceptual overview and detailed explanation of PSO algorithm as well as shows how it can be used for solving ELD problems.

Inherent shortcoming of the traditional methods of finding the ELD is discussed along with other evolutionary methods of finding ELD. A comparative study of different evolutionary programming technique is done and it is shown that particle swarm optimization offer a better result with greater repeatability and lesser time.

The feasibility of the proposed method is demonstrated by using a six generator interconnected system having different cost functions and voltage constraints. The results are compared with those obtained by G.A.

1. INTRODUCTION

1.1 OPTIMIZATON

Optimization is the process of making something better. An engineer or scientist conjures up to a new idea and optimization improves on that idea. Optimization consists in trying variations on an initial concept and using the information gathered to improve on the idea.

Optimization problem may be solved, by choosing an algorithm to conduct state space search. The state space depends upon the representation chosen. A "feasible solution" is a candidate solution that is acceptable without further modifications. A global optimum is a candidate solution whose quality is better than or equal to the quality of every other candidate solution. A local optimum is a candidate solution whose quality cannot be improved by any single move.

Finding the minimum of the non-linear function is difficult, typical approaches involve either linear zing the problem in a very confined region or restricting the optimization to a small region.

Many algorithms that solve the optimization problems attempt to minimize an energy function that combines the cost or performance criteria with penalty functions that implements constraints needed to ensure feasibility. Infeasible solutions should have higher energies than feasible solutions and the better of the two feasible solutions should have lower energy.

1.2 CATEGORIES OF OPTIMIZATION

Optimization algorithms are divided into six categories:-

- Trial and error optimization refers to the process of adjusting variables that affect the optimization without knowing much about the process that produces the output.
- One dimensional and multidimensional optimization; one-dimensional caries only one variable whereas multidimensional has more than one variable.
- Dynamic optimization and Static optimization; in case of dynamic optimization output is a function of time. In case of static output is not a function of time.
- Discrete and Continuous optimization; Discrete optimization has only finite number of possible values whereas in Continuous optimization possible values may be infinite.
- Constrained and unconstrained optimization; constrained optimization includes variables, equalities and inequalities into the cost function, unconstrained optimization allows the variables to any values

1.3 NATURAL OPTIMIZATION METHODS

Evolutionary algorithms are iterative and stochastic optimization techniques inspired by concepts from Darwinian evolution theory. An EA simulates an evolutionary process on a population of individuals with the purpose of evolving the best possible approximate solution to the optimization problem at hand. In the simulation cycle, three operations are typically in play; recombination, mutation, and selection. Recombination and mutation create new candidate solutions, whereas selection weeds out the candidates with low fitness, which is evaluated by the objective, function. Figure illustrates the initialization and the iterative cycle in EAs.

Genetic algorithms: John Holland proposed genetic algorithm in 1975. Genetic algorithms are problem-solving programs that try to mimic the way large populations solve problems over a long period of time, through processes such as reproduction, mutation, and natural selection. To emulate the natural phenomenon of evolution, a genetic algorithm program creates a population of candidate solutions to a particular problem, and through a process of random selection and variation, each generation of the program improves upon the quality of the solution. Consequently, genetic algorithms promote the evolution of solutions by using genetically based processes.

Simulated annelealing: The working principle of simulated annealing is borrowed from metallurgy: a piece of metal is heated and then the metal is left to cool slowly. The slow and regular cooling of the metal allows the atoms to slide progressively toward their most stable, minimal energy, positions. Rapid cooling would have "frozen" them in whatever position they happened to be at that time. The resulting structure of the metal is stronger and more stable.

Ant colony optimization: Ant colony optimization was proposed by Dorigo and Maria in 1997Ant colony optimization (ACO) is a population-based meta-heuristic that can be used to find approximate solutions to difficult optimization problems. When real-world ant colonies are given access to a food source that has multiple approach paths, most ants end up using the shortest and most efficient route. To expedite this process, some ant species deposit a chemical substance called pheromone on the ground when traveling from the nest to the food source. While the process iterates, pheromones are deposited at a higher rate on the shorter paths than the longer ones. When the other ants arrive at a decision point, like an intersection between various paths, they make a probabilistic choice based on the amount of pheromones they smell. After several trips, nearly all of the ants are using the shortest path due to the high concentration of pheromones deposited.

Particle swarm optimization: Particle Swarm Optimization is a biologically inspired method of search and optimization developed in 1995 by Dr. Eberhart and Dr. Kennedy. Based on the social behaviors of birds flocking or fish schooling, this technique represents possible solutions as "particles" as they "fly" like a swarm through the solution space. Like a flock, the swarm gravitates towards the "leader", the current best-known solution, accelerating and turning as better solutions are found. Research on these systems has demonstrated that PSO can efficiently find better solutions than many other techniques for many complex problems.

2. LITERATURE REVIEW

Most power system optimization problems including economic load dispatch have complex and nonlinear characteristics with heavy equality and inequality Constraints. To solve these problems, various salient mathematical approaches have been suggested for the past decades. As an alternative to the conventional mathematical approaches, the heuristic optimization techniques such as genetic algorithms, Tabu search, simulated annealing, and recently introduced particle swarm optimization (PSO) are considered as realistic and powerful solution schemes to obtain the global or quasi-global optimums in power system optimization problems [1].

The primary objectives of this proposed work is to economically schedule the power plants to minimize the system losses, and to demonstrate the potential of the proposed algorithms on real-world problems of system identification and control [2]. The fundamental challenges are: Fitness function design, methods for parameter control, and techniques for optimization. Furthermore, particle swarm optimization problems were studied in the context of the three fundamental challenges. Fitness function design is of major importance for the optimization algorithms, because the fitness function essentially determines how hard the problem is to optimize [3]. There are several sub-aspects of fitness function design. The smoothness of the function is one of primary concern, because a too rugged fitness landscape may disrupt the search and trap the algorithm in a local optimum [4].

Recently, Eberhart and Kennedy suggested a particle swarm optimization (PSO) based on the analogy of swarm of bird and school of fish [5]. The PSO mimics the behaviors of individuals in a swarm to maximize the survival of the species. In PSO, each individual decides his decision using his own experience as well-as other individual’s experiences [6]. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of moving points in a multidimensional space. The individual particles are drawn stochastically toward the position of present velocity of each individual, their own previous best performance, and the best previous performance of their neighbors [7]. The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques [8].

Recently, PSO have been successfully applied to various fields of power system optimization such as power system stabilizer design [9], reactive power and voltage control [10], and dynamic security border identification [11], etc. Besides these power related problems it is a highly reliable tool to solve the global phenomenon's that affects our climate etc. for this specifically, it can be used to identify the possible trajectories of the hurricane and the load forecasting. The original PSO mechanism is directly applicable to the problems with continuous domain and without any constraints [12]. Therefore, it is necessary to revise the original PSO to reflect the equality line quality constraints of the variables in the process of modifying each individual's search. Yoshida et al. [13] suggested a modified PSO to control reactive power and voltage considering voltage security assessment. Since the problem was a mixed-integer nonlinear optimization problem with inequality constraints, they applied the classical penalty method to reflect the constraint-violating variables. Abido [14] developed a revised PSO for determining the optimal values of parameters for power system stabilizers. In practice, an ELD problem is represented as a non-smooth optimization problem with equality and inequality constraints, which makes it difficult to obtain the global optimum. To solve the problems, many salient methods have been proposed such as a mathematical approach [15], dynamic programming [16], improved evolutionary programming [17], neural network approaches [18], [19], and genetic algorithm [20]. We propose an alternative approach to the non-smooth ELD problems using PSO focused on the treatment of the equality and inequality constraints in the process of modifying each individual's search. The inequality constraints in creating initial individuals are easily handled [21, 22]. However, the next position of an individual produced by the PSO algorithm can violate the inequality constraint. In this case, the position of any individual violating the constraints is set to its minimum or maximum position depending on the velocity evaluated [23-26]. The feasibility of the proposed PSO for ELD problems with quadratic and piecewise quadratic cost functions is demonstrated and compared with the existing approaches [27-30].

3. PROBLEM FORMULATION

The work employs particle swarm optimization algorithm for ELD. The work introduces a conceptual overview and detailed explanation of PSO algorithm as well as shows how it can be used for solving ELD problems. Here PSO offer a better result with greater repeatability results in minimization of cost and time.

Finding the minimum of non-linear function is difficult, typical approaches involve either linearization the problem in a very confined region or restricting the optimization to a small region. A comparative study of different evolutionary programming technique is done and it is shown that the particle swarm optimization offer a better result.

A. Objective Function

The objective of ED is to simultaneously minimize the generation cost rate and to meet the load demand of a power system over some appropriate period while satisfying various constraints. To combine the above two constraints into an ED problem, the constrained optimization problem at specific operating interval can be modified as:

$$\text{Min } F_t = \sum_{i=1}^{m} F_i(P_i) = \sum_{i=1}^{m} \alpha_i + \beta_i P_i + \gamma_i P_i^2 \qquad (1)$$

B. Constraints

i) Power Balance

$$\sum_{i=1}^{m} P_i = P_D + P_L, i = l, \ldots\ldots\ldots m. \qquad (2)$$

ii) Generator operation constraints

$$P_i^{\min} \le P_i \le P_i^{\max} (i = l, \ldots\ldots\ldots m). \qquad (3)$$

iii) Line flow Constraints

$$\left| P_{Lf,k} \right| \le P_{Lf,k}^{\max}, k = l, \ldots\ldots\ldots\ldots L \qquad (4)$$

Where the generation cost function $F_i(P_i)$ is usually expressed as quadratic polynomial; α_i, β_i and γ_i are the cost coefficients of the $i-$th generator; m is the number of generators committed to the operating system; P_i is the power output of the $i-$th generator; $P_{Lf,k}$ is the real power flow of the line $j;k$ is the number of transmission lines; and the total transmission network losses is a function of unit power outputs that can be represented using B- coefficients.

$$P_L = B_{00} + \sum_{i=1}^{m} B_{0i} P_i + \sum_{i=1}^{m} \sum_{j=1}^{m} P_i B_{ij} P_j MW \qquad (5)$$

4. OBJECTIVE

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995 [6], inspired by social behavior of bird flocking or fish schooling. This algorithm is initialized with a population of random solutions, called particles. Each particle in PSO flies through the search space with a velocity that is dynamically adjusted according to its own and its companion's historical behaviors. The particles have a tendency to fly toward better search areas over the course of a search process. During flight, each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called lbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest. The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations.

Each particle tries to modify its position using the concept of velocity. The velocity of each agent can be updated by the following equation:

$$v_i^{k+1} = \omega v_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest_i - s_i^k)$$
(6)

where v_i^{k+1} is velocity of agent i at iteration k, ω is weighting function, c1 and c2 are weighting factors, rand1 and rand2 are random numbers between 0 and 1, s_i^k is current position of agent i at iteration k, pbesti is the pbest of agent i, and gbest is the best value so far in the group among the pbests of all agents.

The following weighting function is usually used:

$$\omega = \omega_{\max} - ((\omega_{\max} - \omega_{\min})/(iter_{\max})) \times iter \quad (7)$$

Where, ω max is the initial weight, ω min is the final weight, itermax is the maximum iteration number, and iter is the current iteration number. Using the previous equations, a certain velocity, which gradually brings the agents close to pbest and gbest, can be calculated. The current position (search point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (8)$$

The model using (1) is called Gbest model. The model using (2) in (1) is called inertia weights approach (IWA).

The end criteria are usually one of the following:

- **Maximum number of iterations**: the optimization process is terminated after a fixed number of iterations. Number of iterations without improvement: the optimization process is terminated after some fixed number of iterations if any improvement is not obtained.
- **Minimum objective function error:** the optimization process is terminated if the error between the obtained objective function value and the best fitness value is less than a prefixed anticipated threshold.

5. METHODOLOGY

The concept of PSO can be best understood by the help of the example similar to the one, which led to its development. Imagine a swarm of bees in a field. Their goal is to find in the field the location with the highest density of flowers. Without any knowledge of the field a priori, the bees begin in random locations with random velocities looking for flowers. Each bee can remember the locations that it found the most flowers, and somehow knows the locations where the other bees found an abundance of flowers. Torn between returning to the location where it had personally found the most flowers, or exploring the location reported by others to have the most flowers, the ambivalent bee accelerates in both directions altering its trajectory to fly somewhere between the two points depending on whether nostalgia or social influence dominates its decision Fig 2 (a). Along the way, a bee might find a place with a higher concentration of flowers than it had found previously. It would then be drawn to this new location as well as the location of the most flowers found by the whole swarm. Occasionally, one bee may fly over a place with more flowers than had been encountered by any bee in the swarm. The whole swarm would then be drawn toward that location in additional to their own personal discovery. In this way the bees explore the field: over-flying locations of greatest concentration, then being pulled back toward them. Constantly, they are checking the territory they fly over against previously encountered locations of highest concentration hoping to find the absolute highest concentration of flowers. Eventually, the bees' flight leads them to the one place in the field with the highest concentration of flowers. Soon, all the bees swarm around this point. Unable to find any points of higher flower concentration, they are continually drawn back to the highest flower concentration Fig.2 (b).

5.1 Why PSO?

In general, most real-world optimization problems have several challenging properties. Nearly all problems have a significant number of local optima, and the search space can be so huge that the exact global optimum cannot be found in reasonable time. Additionally, the problems may have multiple conflicting objectives that should be considered simultaneously (e.g., cost versus quality). Moreover, there may be a number of non-linear constraints to be fulfilled by the final solution. For this reason, these algorithms often stagnate at a local Optimum, which makes local search less desirable for many real-world problems. Valuable alternatives are stochastic search methods such as simulated annealing, ant-colony, and evolutionary algorithms. Among these techniques, PSO seem to be a particularly promising approach for several reasons. PSO algorithms are very general regarding the problem types they can be applied to (continuous, mixed-integer etc.). Furthermore, particle swarm can handle problems with any combination of the above-mentioned challenges in real-world problems (local optima, multiple objectives, constraints, and dynamic components).

Naturally, PSO do also have some disadvantages. Unfortunately, they are rather computationally demanding, since many candidate solutions have to be evaluated in the optimization process. However, there has been a recent increase in interest in dealing with this problem and some techniques have been suggested. Furthermore, PSO should not be applied blind-folded to any problem. As mentioned, many simpler and faster techniques exist and they should typically be tried first. In this context, PSO offer the possibility to further improve solutions found by simpler techniques, which can be done by incorporating them in the start population. In addition, PSO typically have a few algorithmic parameters to tune.

6. CONCLUSION

A modified adaptive particle swarm optimization algorithm based on fuzzy and adaptive programming of multi-optimum can be used in conjunction with modified particle swarm optimization algorithm to solve the optimal power flow (OPF) problem.

The modified particle swarm optimization algorithm will help the particles to learn not only from itself and the best one but also from the other individuals in the swarm. By this enhanced study behavior the opportunity to find the global optimum is increased and the influence of initial position of the particle is decreased.

If the proportion factor of multi-optimum programming cannot be dynamically adjusted in the optimization process, the performance of the algorithm will be limited.

The modified adaptive particle swarm optimization algorithm based on fuzzy and adaptive programming of multi-optimum will help to adjust dynamically the proportion factor of multi-optimum programming in the optimization process.

The searching process of the particle swarm optimization is a non-linear and dynamic process. Therefore, when the environment itself is dynamically changed over the time, the algorithm should be able to adapt dynamically to the changing environment. Although the static multi-optimum programming mode improves the general convergence performance of the algorithm compared to the basic PSO, the programming coefficient cannot be adjusted dynamically to the current optimization ability, and getting better programming coefficient need plenty of experiment; therefore its adaptive ability and general convergence performance is limited to some extent.

The idea of a modified adaptive particle swarm optimization algorithm based on fuzzy theory introduces an intelligent method on the basis of multi-optimum programming to program the relationship between the multi-optimum information dynamically in computation process, and then adjust the programming strategies adaptively according to the current optimization ability, which harmonizes the movement relation between itself and the swarm more flexibly, and greatly improves the general convergence performance of the algorithm.

7. ACKNOWLEDGMENTS

Our thanks to the parents who motivated us and Experts who have contributed towards development of this issue.

8. REFERENCES

1. K. Y. Lee and M. A. El-Sharkawi (Editors), Modern *Heuristic Optimization Techniques with Applications* to *Power* Systems, IEEE Power Engineering Society (02TP160), 2W2.

2. J.W.Lamont and E.V.Obessis, “Emission Dispatch Models and Algorithms For The 1990’s.” *IEEE Transaction on Power* Systems.Vol.10, No.2.pp 941- 947,May 1995.

3. K.Srikrishna and C.Palanichamy, “Economic Thermal Power Dispatch with Emission Constraint”, *Journal of the Indian Institute of Engineers (India).* Vol.72, p11, April 1991.

4. E. Ozcan and C. K. Mohan *et al.*, “Particle swarm optimization: surfing the waves,” in *Proc. 1999 Congr. Evolutionary Computation*, Washington, DC, 1999.

5. J. Kennedy and R. Eberhart, “Particle swarm optimization.” *Proceedings on IEEE International Conference on Neural Networks ICNN95).* Vol. IV, pp. 1942-1948, Perth. Australia. 1995.

6. R. C. Eberhart and Y. Shi, “Evolving artificial neural networks,” in *Proc 1998 Int. Conf. Neural Networks and Brain*, Beijing, P.R.C., 1998.

7. E.S. Peer, F den Bergh, A.P. Englebrecht, using “Neighborhoods with the guaranteed convergence PSO”, 2003, IEEE, pp. 235-242.

8. P. J. Angeline, “Using selection to improve particle swarm optimization,” *Proc. IEEE International Conference on Evolutionary Computation*, pp. 84–89, May 1998.

9. M. A. Abido. "Optimal design of power-system stabilizers using particle swarm optimization.”IEEE Trans. *on Energy Conversion*, Vol. 17. No. *3*, pp. 406-413, September 2002

10. S. Naka, T. Genji, T. Yura, and Y. Fukuyama, “Practical distribution state estimation using hybrid particle swarm optimization,” *Proc. IEEE Power Eng. Soc. Winter Meeting*, vol. 2, pp. 815–820, 2001.

11. N. Kassahalidis. M. A. El-Sharkawi. R. J. Marks. L. S. Moulin, and A. P. Alvesdee Silva. "Dynamic security border identification wing enhanced particle swarm optimization," *IEEE Trans. on Power* Systems, Vol. 17, No. 3, pp. 723-729, August 2002.

12. P.H. Chen and H.C. Chang, “Large-Scale economic dispatch by genetic algorithm,” *IEEE Trans. Power System*, vol. 10, pp. 1919–1926, Nov. 1995.

13. H. Yoshida, K. Kawaia. Y. Fukuyanw, S. Takayama, and Y. Nakanishi, "A panicle swarm optimization for reactive power and voltage control

considering voltage security assessment." *IEEE* Trans. *on Power* Systems. Vol. 15, No. 4. pp. 1232-1239, November 2000.

14. D. C.Walters and G. B. Sheble, “Genetic algorithm solution of economic dispatch with valve point loading,” *IEEE Trans. Power Syst.*, vol. 8, pp. 1325–1332, Aug. 1993.

15. C. E. Lin and G. L. Viviani. "Hierarchical economic dispatch for piecewise quadratic cost functions." *IEEE* Trans. *on PAS*, Vol. PAS-103, No. 6, June 1984.

16. I. Wood. And B. F. Wollenhg. *Power Generation. Operation and Control.* John Wile" & Sons. Inc. 1984.

17. Y. M. Park 1. R. Won and J. B. Park. "A new approach to economic load dispatch based on improved evolutionary programming," *Engineering Intelligent* Systems for *Electrical Engineering and Communications.* Vol. 6 No.2, pp. 103-110, June 1998.

18. J. H. Park, Y. *S.* Kim, I. K.Eom and K. Y. Lee, "Economic load dispatch for piecewise quadratic cos1 function using Hop-field neural network." IEEE *Trans. on Power* Systems, Vol. *8,* No. *3,* pp. 1030-1038. August 1993.

19. K. Y. Lee, A. Sode-Yome and J. H. Park "Adaptive Hop-field neural network for economic load dispatch," *IEEE* Trans. *on Power Systems,* Vol. 13. No. 2, pp. 519-526, May 1998.

20. D. C. Wallers and G. B. Sheble. "Genetic algorithm solution of economic dispatch with the valve point loading," *IEEE* Trans. *on Power Systems.* Vol. X.No.3.p~. 1325-1332.August 1993.

21. H. Saadat, *Power System Analysis*. New York: McGraw-Hill, 1999.

22. G.Ciuprina, D. loan. And I. Munteanu, “use of intelligent-panicle swarm optimization in electromagnetic” *IEEE* Trans. on *Magnetics,* Vol. *38,* No. 2, pp 1037-1040, March 2002.

23. Kennedy, J., Eberhart, R. C., and Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers.

24. Krink, T., Ursem, R. K., and Filipiˇc, B. (2001). Evolutionary Algorithms in Control Optimization: The Greenhouse Problem. In Spector et al., editors, *Proceedings of the Third Genetic and Evolutionary Computation Conference (GECCO-2001),* volume 1, pages 440–447.

25. Clerc, M. (1999). The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. In Angeline et al., editors, *Proceedings of the Congress of Evolutionary Computation (CEC-1999),* volume 3, pages 1951–1957.

26. Davis, editor (1987). *Genetic Algorithms and Simulated Annealing*. Research Notes in Artificial Intelligence. Pitman Publishing.

27. Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.

28. Eberhart, R. C. and Shi, Y. (2001). Tracking and Optimizing Dynamic Systems with Particle Swarms. In *Proceedings of the Third Congress on Evolutionary Computation (CEC-2001)*, pages 94–100.

29. Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons.

30. Fukuyama, Y. and Yoshida, H. (2001). A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems. In *Proceedings of the 2001 Congress on Evolutionary Computation (CEC-2001)*, pages 87–93.