2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Particle Swarm Optimization with Quasi-Newton Local Search for Solving Economic Dispatch Problem

Leandro dos Santos Coelho and Viviana Cocco Mariani

Abstract— **Particle swarm optimization (PSO) is a population-based swarm intelligence algorithm driven by the simulation of a social psychological metaphor instead of the survival of the fittest individual. Based on the swarm intelligence theory, this paper discusses the use of PSO with a Quasi-Newton (QN) local search method. The PSO is used to produce good potential solutions, and the QN is used to fine-tune of final solution of PSO. The hybrid methodology is validated for a test system consisting of 13 thermal units whose incremental fuel cost function takes into account the valve-point loading effects.**

I. INTRODUCTION

THE objective of the economic dispatch problem (EDP) of electric power generation, whose characteristics are complex and highly nonlinear, is to schedule the committed generating unit outputs so as to meet the required load demand at minimum operating cost while satisfying all unit and system equality and inequality constraints [1].

Recently, as an alternative to the conventional mathematical approaches, modern heuristic optimization techniques such as simulated annealing, evolutionary algorithms, neural networks, ant colony, and taboo search have been given much attention by many researchers due to their ability to find an almost global optimal solution in EDPs [2]-[7].

In this paper, an alternative hybrid method is proposed. The proposed hybrid method combines the particle swarm optimization (PSO) in evolution phase and the Quasi-Newton (QN) technique in the learning phase (after the stopping criterion of PSO be satisfied) to solve the EDP associated with the valve-point effect. The hybrid method of optimization adopted in this paper is also denominated in the literature of the hybrid algorithm, algorithm with local search, memetic algorithm or optimization based in Lamarckian evolution [8], [9].

Manuscript received March 13, 2006.

L. S. Coelho is with the Pontifical Catholic University of Parana, Production and Systems Engineering Graduate Program, Automation and Systems Laboratory, PUCPR/PPGEPS/LAS, Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Parana, Brazil (e-mail: leandro.coelho@pucpr.br).

V. C. Mariani is with the Pontifical Catholic University of Parana, Mechanical Engineering Graduate Program, PUCPR/PPGEM, Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Parana, Brazil (e-mail: viviana.mariani@pucpr.br).

An economic dispatch problem with 13 unit test system using nonsmooth fuel cost function [10] is employed in this paper for demonstrate the performance of the proposed hybrid method. The results obtained with the hybrid approach were analyzed and compared with those obtained in recent literature.

The rest of the paper is organized as follows: section 2 describes the EDP, while section 3 explains the PSO and QN concepts. Section 4 presents the simulation results of the 15 unit test problem optimization and compares methods to solve the case study. Lastly, section 5 outlines our conclusions and future research.

II. DESCRIPTION OF ECONOMIC DISPATCH PROBLEM

The objective of the economic dispatch problem is to minimize the total fuel cost at thermal power plants subjected to the operating constraints of a power system. Therefore, it can be formulated mathematically with an objective function and two constraints. The equality and inequality constraints are represented by equations (1) and (2) given by:

$$\sum_{i=1}^{n} P_i - P_L - P_D = 0 \tag{1}$$

$$P_i^{min} \le P_i \le P_i^{max} \tag{2}$$

In the power balance criterion, an equality constraint must be satisfied, as shown in equation (1). The generated power should be the same as the total load demand plus total line losses. The generating power of each generator should lie between maximum and minimum limits represented by equation (2), where P_i is the power of generator i (in MW); n is the number of generators in the system; P_D is the system's total demand (in MW); P_L represents the total line losses (in MW) and P_i^{min} and P_i^{max} are, respectively, the output of the minimum and maximum operation of the generating unit i (in MW). The total fuel cost function is formulated as follows:

$$min\ f = \sum_{i=1}^{n} F_i(P_i) \tag{3}$$

where F_i is the total fuel cost for the generator unity i (in \$/h), which is defined by equation:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \tag{4}$$

where a_i, b_i and c_i are cost coefficients of generator i.

A cost function is obtained based on the ripple curve for more accurate modeling. This curve contains higher order nonlinearity and discontinuity due to the valve point effect, and should be refined by a sine function. Therefore, equation (4) can be modified [11], as:

$$\widetilde{F}_i(P_i) = F(P_i) + \left| e_i sin\left(f_i\left(P_i^{\min} - P_i\right)\right)\right| \quad \text{or} \tag{5}$$

$$\widetilde{F}_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i sin\left(f_i\left(P_i^{\min} - P_i\right)\right)\right| \tag{6}$$

where e_i and f_i are constants of the valve point effect of generators. Hence, the total fuel cost that must be minimized, according to equation (3), is modified to:

$$min\ f = \sum_{i=1}^{n} \widetilde{F}_i(P_i) \tag{7}$$

where $\widetilde{F}_i$ is the cost function of generator i (in \$/h) defined by equation (6). In the case study presented here, we disregarded the transmission losses, P_L; thus, $P_L = 0$.

III. Optimization Methods of Economic Dispatch Problem

A. Particle Swarm Optimization

The PSO originally developed by Kennedy and Eberhart in 1995 [12], [13] is a population-based swarm algorithm. Swarm intelligence is an emergent research area with populational and evolutionary characteristics similar to those of genetic algorithms. Swarm intelligence is inspired by nature, based on the fact that the individual experience of live animals in a group contributes to the group's overall experience, strengthening it in relation to others. However, swarm intelligence differs insofar as it emphasizes cooperative behavior among group members. Swarm intelligence is used to solve optimization and cooperating problems among intelligent agents.

Similarly to genetic algorithms [14], the PSO is an optimization tool based on a population, where each member is seen as a particle, and each particle is a potential solution to the problem under analysis. Each particle in PSO has a randomized velocity associated to it, which moves through the space of the problem. However, unlike genetic algorithms, PSO does not have operators, such as crossover and mutation. PSO does not implement the survival of the fittest individuals; rather, it implements the simulation of social behavior.

Each particle in PSO keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called *pbest*. Another "*best*" value that is tracked by the *global* version of the particle swarm optimizer is the overall best value and its location obtained so far by any particle in the population. This location is called *gbest*.

The PSO concept consists of, in each time step, changing (accelerating) the velocity of each particle flying toward its *pbest* and *gbest* locations (global version of PSO). Acceleration is weighted by random terms, with separate random numbers being generated for acceleration toward *pbest* and *gbest* locations, respectively. The procedure for implementing the global version of PSO is given by the following steps [15]-[17] (see also the PSO flow chart in Fig. 1):

(i) Initialize a population (array) of particles with random positions and velocities in the n dimensional problem space using a uniform probability distribution function.

(ii) Evaluate the fitness value of each particle.

(iii) Compare each particle's fitness with the particle's *pbest*. If the current value is better than *pbest*, then set the *pbest* value equal to the current value and the *pbest* location equal to the current location in n-dimensional space.

(iv) Compare the fitness with the population's overall previous best. If the current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.

(v) Change the velocity and position of the particle according to equations (8) and (9), respectively:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot ud \cdot [p_i(t) - x_i(t)] + c_2 \cdot Ud \cdot [p_g(t) - x_i(t)] \tag{8}$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \tag{9}$$

where $i=1,2,\ldots,N$ indicates the number of particles of population (swarm); $t=1,2,\ldots t_{max}$, indicates the iterations, w is a parameter called the inertial weight; $v_i = [v_{i1}, v_{i2}, \ldots v_{in}]^T$ stands for the velocity of the i-th particle, $x_i = [x_{i1}, x_{i2}, \ldots x_{in}]^T$ stands for the position of the i-th particle of population, and $p_i = [p_{i1}, p_{i2}, \ldots p_{in}]^T$ represents the best previous position of the i-th particle. Positive constants c_1 and c_2 are the cognitive and social components, respectively, which are the acceleration constants responsible for varying the particle speed towards *pbest* and *gbest*, respectively. Index g represents the index of the best particle among all the particles in the swarm. Variables ud and Ud are two random functions in the range [0,1]. Equation (1) represents the position update, according to its previous position and its velocity, considering $\Delta t = 1$.

(iv) Loop to step (ii) until a stop criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

Positive constants c_1 and c_2 are called cognitive and social components, respectively. These are the acceleration constants responsible for varying the particle velocity towards *pbest* and *gbest*. Particle velocities in each dimension are clamped to a maximum velocity V_{max}. If the sum of accelerations causes the velocity in that dimension to exceed V_{max}, which is a parameter specified by the user, then the velocity in that dimension is limited to V_{max}.

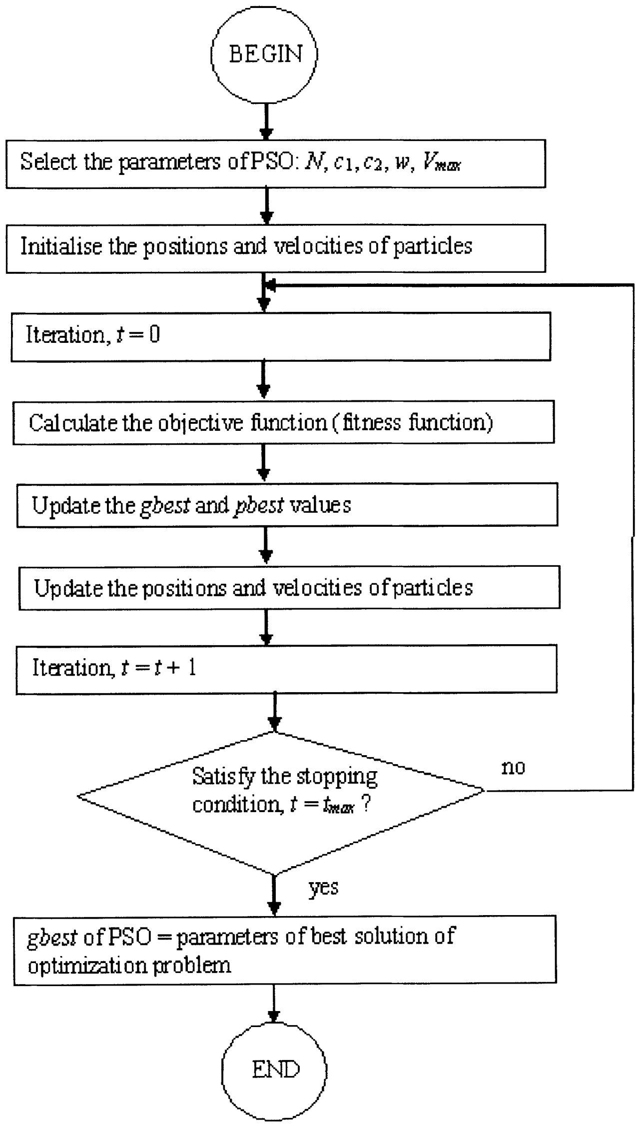

Fig. 1. Flow chart in PSO approach.

V_{max} is a parameter serving to determine the resolution with which the regions around the current solutions are searched. If V_{max} is too high, the PSO facilitates a global search, and particles might fly past good solutions. Conversely, if V_{max} is too small, the PSO facilitates a local search and particles may not explore sufficiently beyond locally good regions.

Previous experience with PSO (trial and error, mostly) led us to set the V_{max} to 20% of the dynamic range of the variable in each dimension.

The first part in equation (6) is the momentum part of the particle. The inertia weight w represents the degree of the momentum of the particles. The second part is the 'cognition' part, which represents the independent thinking of the particle itself.

B. Combining of PSO with Quasi-Newton method

A direct application of Newton's method would be computationally prohibitive due to the computational cost of the evaluation of the Hessian matrix. Alternative approaches, known as QN or variable metric methods, build an approximation of the inverse of the Hessian using only information of the first derivatives of the error function over a number of steps [18].

The two most commonly used update formulae are the Daivdson-Fletcher-Power (DFP) and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) procedures. In this work, the BFGS approach was used. The BFGS routine in this paper is the one provided by the Matlab Optimization Toolbox (*fminunc* rotine). Details of the BFGS procedure are presented in Fletcher [19]-[21].

QN method and PSO have advantages that complement each other. The proposed combination of PSO with QN for local search consists of a form of sequential hybridization based on [2], [8] and [9]. Basically, in this combined method, the PSO is applied to the optimization problem and the best solution (or other chosen solution) obtained by PSO is used as starting point for the QN method.

IV. CASE STUDY OF 13 THERMAL UNITS

This case study consisted of 13 thermal units of generation with the effects of valve-point loading, as given in Appendix (Table I). The data shown in Table I is also available in [10] and [22]. In this case, the load demand expected to be determined was $P_D = 1800$ MW.

The results obtained for case study I are given in Table II, which shows that the PSO-QN succeeded in finding the best solution for the tested methods. However, the QN outperformed the other tested methods in terms of solution time. The best results obtained for solution vector P_i, i=1,..,13 with PSO-QN with minimum cost of 17979.6490 \$/h is given in Table III.

Table IV compares the results obtained in this paper with those of other studies reported in the literature. Note that in studied case, the result reported here using PSO-QN is comparatively lower than recent studies presented in the literature, except to hybrid particle swarm with SQP proposed in [1].

TABLE I
DATA FOR THE 13 THERMAL UNITS

G	P_i^{min}	P_i^{max}	a	b	c	e	f
1	0	680	0.00028	8.10	550	300	0.035
2	0	360	0.00056	8.10	309	200	0.042
3	0	360	0.00056	8.10	307	150	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.60	126	100	0.084
11	40	120	0.00284	8.60	126	100	0.084
12	55	120	0.00284	8.60	126	100	0.084
13	55	120	0.00284	8.60	126	100	0.084

TABLE II
CONVERGENCE RESULTS (50 RUNS) OF A CASE STUDY OF 13 GENERATING UNITS WITH VALVE POINT AND $P_D = 1800$ MW.

Method	Mean Time (s)	Minimum Cost ($/h)	Mean Cost ($/h)	Standard Deviation of Cost ($/h)	Maximum Cost ($/h)
QN	0.32	17995.0995	18276.3421	145.1728	18631.2543
PSO	2.92	17996.7565	18087.1377	54.8552	18144.5428
PSO-QN	2.99	**17979.6490**	**18015.1716**	**46.8181**	**18058.7126**

TABLE III
BEST RESULT (50 RUNS) OBTAINED FOR THE CASE STUDY I USING DEC(1)-SQP(1).

Power	Generation	Power	Generation
P_1	486.6072	P_8	87.6359
P_2	246.1276	P_9	90.6222
P_3	223.4429	P_{10}	40.0000
P_4	83.9027	P_{11}	40.0000
P_5	132.6469	P_{12}	89.2177
P_6	142.3588	P_{13}	67.3884
P_7	60.0000		

TABLE IV
COMPARISON OF CASE STUDY RESULTS FOR FUEL COSTS PRESENTED IN THE LITERATURE.

Optimization Technique	Case Study with 13 Thermal Units
Evolutionary programming [10]	17994.07
Particle swarm optimization [1]	18030.72
Hybrid evolutionary programming with SQP [1]	17991.03
Hybrid particle swarm with SQP [1]	17969.93
Best result of this paper	**17979.6490** (using PSO-QN)

V. Conclusion and Future Research

This paper discusses the use of PSO with a Quasi-Newton (QN) local search method. The hybrid methodology was successfully validated for a test system consisting of 13 thermal units whose incremental fuel cost function takes into account the valve-point loading effects.

Future work will include the hybridization of the PSO and other local search methods, such as branch-and-bound and simulated annealing. This approach, combining of the efficient global search of PSO and the effectiveness of deterministic local search, possibly will give good results for EDPs.

References

[1] T. A. A. Victoire and A. E. Jeyakumar, "Hybrid PSO-SQP for economic dispatch with valve-point effect," *Electric Power Systems Research*, vol. 71, no. 1, pp. 51-59, 2004.

[2] L. S. Coelho and V. C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Transactions on Power Systems*, 2006 (accepted for publication in issue of May 2006).

[3] T. Yalcinoz and H. Altun, "Power economic dispatch using a hybrid genetic algorithm," *IEEE Power Engineering Review*, March, pp. 59-60, 2001.

[4] Y. H. Song and C. S. V. Chou, "Large-scale economic dispatch by artificial ant colony search algorithms," *Electric Machines and Power Systems*, vol. 27, pp. 679-690, 1999.

[5] W. -M. Lin, F. -S. Cheng, and M. -T. Tsay, "An improved tabu search for economic dispatch with multiple minima," *IEEE Transactions on Power Systems*, vol. 17, no. 1, pp. 108-112, 2002.

[6] M. Basu, "A simulated annealing-based goal-attainment method for economic emission load dispatch of fixed head hydrothermal power systems," *International Journal of Electrical Power & Energy Systems*, vol. 27, no. 2, pp. 147-153, 2005.

[7] S. Balakrishnan, P. S. Kannan, C. Aravindan and P. Subathra, "On-line emission and economic load dispatch using adaptive Hopfield neural network," *Applied Soft Computing*, vol. 2, no. 4, pp. 297-305, 2003.

[8] Ph. Preux and E. -G. Talbi, "Towards hybrid evolutionary algorithms," *International Transactions in Operational Research*, vol. 6, pp. 557-570, 1999.

[9] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms; model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474-488, 2005.

[10] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 83-94.

[11] A. J. Wood and B. F. Wollenberg, *Power generation, operation and control*, New York, John Wiley & Sons, 1994.

[12] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proccedings of the 6th. Int. Symposium on Micro Machine and Human Science*, Nagoya, Japan, Piscataway, NJ: IEEE Service Center, pp. 39-43, 1995.

[13] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings of the IEEE International Conference on Neural Networks IV*, Piscataway, NJ: IEEE Service Center, pp. 1942-1948, 1995.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley: Reading, MA, 1989.

[15] R. A. Krohling, L. S. Coelho and Y. Shi, "Cooperative particle swarm optimization for robust control system design." *7th World Conference on Soft Computing in Industrial Applications*, Granada, Spain, On-line, *http://site: decsai.ugr.es/wsc7*, 2002.

[16] R. A. Krohling, F. Hoffmann, and L. S. Coelho, "Co-evolutionary particle swarm optimization for min-max problems using Gaussian distribution," *Proceedings of the Congress on Evolutionary Computation*, Portland, Oregon USA, pp. 959-964, 2004.

[17] L. S. Coelho and R. A. Krohling, "Predictive controller tuning using modified particle swarm optimization based on Cauchy and Gaussian distribution," in *Soft Computing: Methodologies and Applications*, Advances in Soft Computing Series, F. Hoffmann, M. Köppen, F. Klawonn, R. Roy (editors), Springer-Verlag, UK, 2005.

[18] C. M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, New York, USA, 1995.

[19] R. Fletcher, *Practical methods of optimization*, 2nd edition, John Wiley & Sons, New York, NY, 1987.

[20] M. S. Bazarra, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*, John Wiley & Sons, New York, USA, 2nd edition, 1979.

[21] J. Nocedal, "Theory of algorithms for unconstrained optimization," *Acta Numerica*, vol. 1, pp. 199-242, 1991.

[22] K. P. Wong and Y. W. Wong, "Genetic and genetic/simulated-annealing approaches to economic dispatch," *IEE Proc. Control, Generation, Transmission and Distribution*, vol. 141, no. 5, pp. 507-513, 1994.