

A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch

Leandro dos Santos Coelho ^{a,*}, Viviana Cocco Mariani ^b

^a *Industrial and Systems Engineering Program, PPGEPS, Pontifical Catholic University of Paraná, PUCPR, Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil*

^b *Mechanical Engineering Graduate Program, PPGEM, Pontifical Catholic University of Paraná, PUCPR, Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil*

Accepted 2 January 2007

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 chaotic systems theory, this paper proposed a novel chaotic PSO combined with an implicit filtering (IF) local search method to solve economic dispatch problems. Since chaotic mapping enjoys certainty, ergodicity and the stochastic property, the proposed PSO introduces chaos mapping using Hénon map sequences which increases its convergence rate and resulting precision. The chaotic PSO approach is used to produce good potential solutions, and the IF 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. Simulation results are promising and show the effectiveness of the proposed approach.

© 2009 Published by Elsevier Ltd.

1. 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, artificial neural network, 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]. One of these modern heuristic optimization paradigms is the Particle Swarm Optimization (PSO) [8–10].

* Corresponding author.

E-mail addresses: leandro.coelho@pucpr.br (Leandro dos Santos Coelho), viviana.mariani@pucpr.br (V.C. Mariani).

PSO is a kind of evolutionary algorithm based on a population of potential solutions and motivated by the simulation of social behavior instead of the survival of the fittest individual. It is a population-based evolutionary algorithm. Similar to the other population-based evolutionary algorithms, PSO is initialized with a population of random solutions. Unlike the most of the evolutionary algorithms, each potential solution (individual) in PSO is also associated with a randomized velocity, and the potential solutions, called particles, are then “flowed” through the problem space.

The approach of composite configuration by deterministic techniques combined with PSO algorithms is a promising alternative in optimization and must be evaluated. In this paper, an alternative hybrid method is proposed. The proposed hybrid method combines the PSO using chaotic sequences generate by Hénon map in evolution phase and the implicit filtering (IF) algorithm in the learning phase (after the stopping criterion of chaotic PSO be satisfied) to solve the EDP associated with the valve-point effect. The IF algorithm is a projected quasi-Newton method that uses finite difference gradients. The difference increment is reduced as the optimization progresses, thereby avoiding some local minima, discontinuities, or nonsmooth regions that would trap a conventional gradient-based method. 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 [11,12].

Chaos describes the complex behavior of a nonlinear deterministic system. Optimization algorithms based on the chaos theory are search methodologies that differ from any of the existing traditional stochastic optimization techniques. Due to the non-repetition of chaos, it can carry out overall searches at higher speeds than stochastic ergodic searches that depend on probabilities. In this context, the literature contains several optimization algorithms using chaotic sequences for solving design problems, such as the presented works in [13–21]. The application of chaotic sequences instead of random sequences in PSO is a powerful strategy to diversify the population of particles and improve the PSO’s performance in preventing premature convergence to local minima.

An EDP problem with 13 unit test system using nonsmooth fuel cost function [22] is employed in this paper for demonstrate the performance of the proposed hybrid method. The results obtained with the chaotic PSO approach using Hénon map and an IF local search 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, chaotic PSO and IF concepts. Section 4 presents the simulation results of the 13 unit test problem optimization and compares methods to solve the case study. Lastly, Section 5 outlines our conclusions and future research.

2. 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 Eqs. (1) and (2) given by:

$$\sum_{i=1}^n P_i - P_L - P_D = 0 \quad (1)$$

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (2)$$

In the power balance criterion, an equality constraint must be satisfied, as shown in Eq. (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 Eq. (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:

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

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

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \quad (4)$$

where a_i , b_i and c_i are 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, Eq. (4) can be modified [23], as:

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

$$\tilde{F}_i(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \sin(f_i(P_i^{\min} - P_i))| \quad (6)$$

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 Eq. (3), is modified to:

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

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

3. Optimization methods to solve the economic dispatch problem

3.1. Particle swarm optimization

The PSO originally developed by Kennedy and Eberhart in 1995 [8,9] 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 [24], 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 problem space. 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. In this work, the *gbest* version of PSO is adopted. The *gbest* (star) version is a fully connected neighborhood relation. Each particle has all the other particles as neighbors; this implies that the global best particle-position for all particles is identical [25] (see Fig. 1).

The procedure for implementing the global version of PSO is given by the following steps [26,27] (see also the PSO flow chart in Fig. 2):

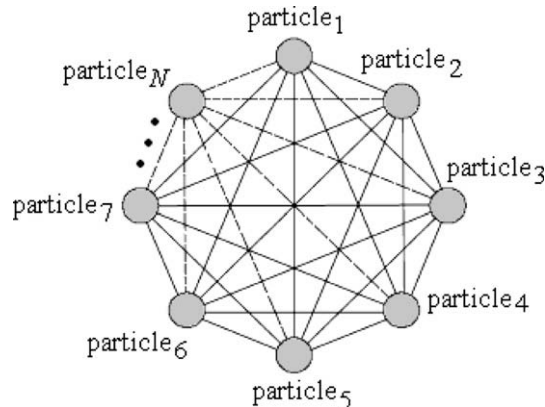


Fig. 1. PSO using *gbest* neighborhood topology.

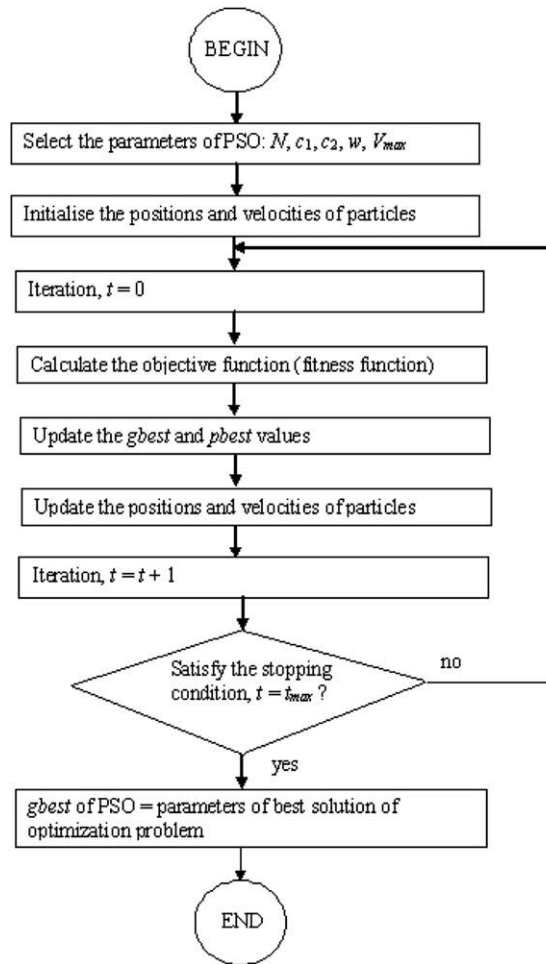


Fig. 2. Flow chart in PSO approach.

- (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 Eqs. (8) and (9), respectively:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot ud_{i,j}(t) \cdot [p_i(t) - x_i(t)] + c_2 \cdot Ud_{i,j}(t) \cdot [p_g(t) - x_i(t)] \quad (8)$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \quad (9)$$

where $i = 1, 2, \dots, N$ indicates the number of particles of population (swarm); $t = 1, 2, \dots, t_{\max}$, indicates the iterations, w is a parameter called the inertia weight; $v_i = [v_{i1} v_{i2} v_{in}]^T$ stands for the velocity of the i -th particle, $x_i = [x_{i1} x_{i2}, \dots, x_{in}]^T$ stands for the position of the i -th particle of population, and $p_i = [p_{i1} p_{i2}, \dots, p_{in}]^T$ 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 velocity towards $pbest$ and $gbest$, respectively. Index g represents the index of the best particle among all the particles in the swarm. Variables $ud_{i,j}(t)$ and $Ud_{i,j}(t)$ are uniformly distributed random numbers in the range $[0, 1]$ of the j -th design variable of i -th particle. Eq. (9) 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).

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} .

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 particle in each dimension.

The first part in Eq. (8) 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.

3.2. Chaotic particle swarm optimization

In PSO design, the concepts of optimization based on chaotic sequences can be a good alternative to provide diversity in populations of PSO approaches. Different types of equations have been considered in literature for applications in optimization methods. The logistic equation and other equations, such as ten map, Gauss map, Lozi map, sinusoidal iterator, Chua’s oscillator, Mackey–Glass system, Lorenz system, Ikeda map, and others, have been adopted instead of random ones and very interesting results [2,13–21].

The parameters $ud_{i,j}(t)$ and $Ud_{i,j}(t)$ are important control parameters that affect the PSO’s convergence. This paper provides new approaches introducing chaotic mapping with ergodicity, irregularity and the stochastic property in PSO to improve the global convergence in substitution of parameters $ud_{i,j}(t)$ and $Ud_{i,j}(t)$. The use of chaotic sequences in PSO can be helpful to escape more easily from local minima than the traditional PSO methods.

New PSO approaches are proposed here based on Hénon map [28]. Hénon introduced this map as a simplified version of the Poincaré map of the Lorenz system. The Hénon equations are given by

$$y(t) = 1 - a \cdot y(t-1) + z(t-1) \quad (10)$$

$$z(t) = b \cdot y(t-1) \quad (11)$$

For $a = 1.4$ and $b = 0.3$ (the values for which the Hénon map has a strange attractor), the Hénon map is used in this work, as presented in Fig. 3. In this case, the output values of $z(t) \in [-0.3854, 0.3819]$. In this work, the values of $z(t)$ are normalized in the range $[0, 1]$. Another Hénon map using the same Eqs. (10) and (11) is used to generate the variable $h(t)$ normalized in the range $[0, 1]$.

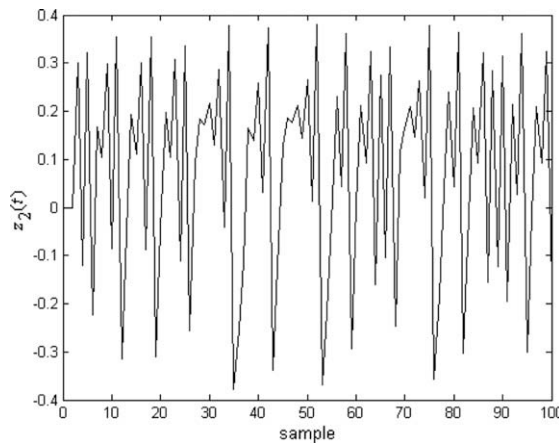


Fig. 3. Example of evolution (100 samples) of Hénon map for $a = 1.4$ and $b = 0.3$.

These new PSO approaches combined with chaotic sequences based on Hénon map (HPSO) are described as follows:
Approach 1 – HPSO1: Parameter $ud_{i,j}(t)$ of Eq. (8) is modified by the following equation:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot z_{i,j}(t) \cdot [p_i(t) - x_i(t)] + c_2 \cdot Ud_{i,j}(t) \cdot [p_g(t) - x_i(t)] \quad (12)$$

where $z_{i,j}(t)$ is given by Hénon map scaled with values between 0 and 1.

Approach 2 – CPSO2: Parameter $Ud_{i,j}(t)$ of Eq. (8) is modified by the following equation:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot ud_{i,j}(t) \cdot [p_i(t) - x_i(t)] + c_2 \cdot z_{i,j}(t) \cdot [p_g(t) - x_i(t)] \quad (13)$$

Approach 3 – HPSO3: Parameters $ud_{i,j}(t)$ and $Ud_{i,j}(t)$ of Eq. (8) are modified by the following equation:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot z_{i,j}(t) \cdot [p_i(t) - x_i(t)] + c_2 \cdot h_{i,j}(t) \cdot [p_g(t) - x_i(t)] \quad (14)$$

where $z_{i,j}(t)$ and $h_{i,j}(t)$ are given by Hénon map scaled with values between 0 and 1.

3.3. Combining of PSO and HPSO with IF local search

The quasi-Newton implicit filtering algorithms differ from other methods in the literature that use either inaccurate gradient information, only samples of the function, or difference or interpolatory approximations to gradients and/or Hessians.

Implicit filtering, originally proposed in the context of computer aided design of semiconductors [29] is a generalization of the gradient projection algorithm of [30] in which derivatives are computed with difference quotients. The step sizes (called scales) in the difference quotients are changed as the iteration progresses with the goal of avoiding local minima that are caused by high-frequency, low amplitude oscillations. Real filtering could be performed, but this requires sampling and filtering the entire solution space and thus, is computationally quite expensive. Implicit filtering is very similar to adaptive meshing schemes used by the computational fluid mechanics community to avoid unwanted harmonics. The algorithm is fully described in [31,32].

IF method and PSO approaches have advantages that complement each other. The proposed combination of PSO or HPSO with IF for local search consists of a form of sequential hybridization based on [2,11,12]. Basically, in this combined method, the PSO or HPSO is applied to the optimization problem and the best solution (or other chosen solution) obtained by PSO or HPSO is used as starting point for the IF method.

4. Case study of 13 thermal units and analysis of optimization results

This case study consisted of 13 thermal units of generation with the effects of valve-point loading, as given in Appendix (Table 1). The data shown in Table 1 is also available in [22,33]. In this case, the load demand expected to be determined was $P_D = 1800$ MW.

Table 1
Data for the 13 thermal units

Thermal unit	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

Each optimization method was implemented in Matlab (MathWorks). All the programs were run on a 3.2 GHz Pentium IV processor with 2GB of random access memory. In each case study, 50 independent runs were made for each of the optimization methods involving 50 different initial trial solutions for each optimization method. In this paper, the IF routine is adopted using 2000 cost function evaluations in each run.

In this case study, the population size N was 20 and the stopping criterion t_{\max} was 800 generations, and the inertia weight w to linearly decrease from 0.9 to 0.4 for the PSO (using $c_1 = c_2 = 2.05$) and HPSO approaches. A key factor in the application of optimization methods is how the algorithm handles the constraints relating to the problem. In this work, a penalty-based method proposed in [2] was used.

The results obtained for this case study are given in Table 2, which shows that the HPSO-IF succeeded in finding the best solution for the tested methods. However, the IF outperformed the other tested methods in terms of solution time. The best results obtained for solution vector P_i , $i = 1, \dots, 13$ with HPSO1-IF with minimum cost of 17964.6772 is given in Table 3.

Table 4 compares the results obtained in this paper with those of other studies reported in the literature. Note that in studied case, the best result reported here using HPSO-IF is comparatively lower than recent studies presented in the literature.

Table 2

Convergence results (50 runs) of a case study of 13 thermal units with valve-point and $P_D = 1800$ MW

Method	Mean time (s)	Maximum cost (\$/h)	Minimum cost (\$/h)	Mean cost (\$/h)
IF	0.11	19596.3745	18150.4924	18910.6512
PSO	1.69	19329.7864	18798.3258	19077.5215
HPSO1	1.72	19318.1177	18767.4843	19062.7045
HPSO2	1.72	19468.6801	18787.5006	19102.8226
HPSO3	1.73	19538.6992	18706.1454	19067.4146
PSO-IF	1.81	19332.0879	18057.6716	18832.8248
HPSO1-IF	1.83	19396.8603	17964.6772	18816.3809
HPSO2-IF	1.83	19538.0853	18049.4178	18887.0125
HPSO3-IF	1.85	19546.4058	18032.0175	18986.9347

Table 3

Best result (50 runs) obtained for the case study using HPSO1-IF

Power	Generation (MW)	Power	Generation (MW)
P_1	628.3179	P_8	109.8951
P_2	224.3921	P_9	109.8607
P_3	148.1492	P_{10}	40.0000
P_4	109.8661	P_{11}	40.0000
P_5	60.0000	P_{12}	55.0000
P_6	109.8330	P_{13}	55.0000
P_7	109.6859	$\sum_{i=1}^{13} P_i$	1800.0000

Table 4

Comparison of best results for fuel costs presented in the literature

Optimization technique	Case study with 13 thermal units
Evolutionary programming [22]	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	17964.6772 (using HPSO1-IF)

5. Conclusions and future research

This paper discusses the use of HPSO with an IF 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.

From the comparison of the results for the case study through classical PSO and PSO-IF methods, it has been shown that the HPSO1, HPSO3, HPSO1-IF and HPSO2-IF have the ability to search better optimum solution. However, in future works will include a detailed study of metrics, such as computation efficiency and convergence characteristics, of the HPSO and HPSO-IF approaches for other EDPs.

Acknowledgement

This work was supported by the National Council of Scientific and Technologic Development of Brazil – CNPq – under Grant 309646/2006-5/PQ.

References

- [1] Victoire TAA, Jeyakumar AE. Hybrid PSO-SQP for economic dispatch with valve-point effect. *Electr Power Syst Res* 2004;71(1):51–9.
- [2] Coelho LS, Mariani VC. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Trans Power Syst* 2006;21(2):989–96.
- [3] Yalcinoz T, Altun H. Power economic dispatch using a hybrid genetic algorithm. *IEEE Power Eng Rev* 2001;21(3):59–60.
- [4] Song YH, Chou CSV. Large-scale economic dispatch by artificial ant colony search algorithms. *Electr Mach Power Syst* 1999;27(7):679–90.
- [5] Lin WM, Cheng FS, Tsay MT. An improved tabu search for economic dispatch with multiple minima. *IEEE Trans Power Syst* 2002;17(1):108–12.
- [6] Basu M. A simulated annealing-based goal-attainment method for economic emission load dispatch of fixed head hydrothermal power systems. *Int J Electr Power Energy Syst* 2005;27(2):147–53.
- [7] Balakrishnan S, Kannan PS, Aravindan C, Subathra P. On-line emission and economic load dispatch using adaptive Hopfield neural network. *Appl Soft Comput* 2003;2(4):297–305.
- [8] Kennedy JF, Eberhart RC. Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, Perth, Australia; 1995. p. 1942–8.
- [9] Eberhart RC, Kennedy JF. A new optimizer using particle swarm theory. In: *Proceedings of international symposium on micro machine and human science*, Japan; 1995. p. 39–43.
- [10] Kennedy JF, Eberhart RC, Shi Y. *Swarm intelligence*. San Francisco, USA: Morgan Kaufmann Publishers; 2001.
- [11] Preux Ph, Talbi EG. Towards hybrid evolutionary algorithms. *Int Trans Oper Res* 1999;6(6):557–70.
- [12] Krasnogor N, Smith J. A tutorial for competent memetic algorithms; model, taxonomy, and design issues. *IEEE Trans Evolut Comput* 2005;9(5):474–88.
- [13] Li B, Jiang W. Optimizing complex functions by chaos search. *Cybernet Syst* 1998;29(4):409–19.
- [14] Coelho LS, Lopes HS. Supply chain optimization using chaotic differential evolution method. In: *Proceedings of IEEE international conference on systems, man, and cybernetics*, Taipei, Taiwan; 2006.
- [15] Chang WD. Parameter identification of Rössler's chaotic system by an evolutionary algorithm. *Chaos, Solitons & Fractals* 2006;29(5):1047–53.
- [16] Liu B, Wang L, Jin YH, Tang F, Huang DX. Directing orbits of chaotic systems by particle swarm optimization. *Chaos, Solitons & Fractals* 2006;29(2):454–61.
- [17] Zuo XQ, Fan YS. A chaos search immune algorithm with its application to neuro-fuzzy controller design. *Chaos, Solitons & Fractals* 2006;30(1):94–109.
- [18] Liu B, Wang L, Jin YH, Tang F, Huang DX. Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals* 2005;25(5):1261–71.
- [19] Li L, Yang Y, Peng H, Wang X. Parameters identification of chaotic systems via chaotic ant swarm. *Chaos, Solitons & Fractals* 2006;28(5):1204–11.
- [20] Caponetto R, Fortuna L, Fazzino S, Xibilia MG. Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Trans Evolut Comput* 2003;7(3):289–304.
- [21] Zilong G, Sun'an W, Jian Z. A novel immune evolutionary algorithm incorporating chaos optimization. *Pattern Recogn Lett* 2006;27(1):2–8.
- [22] Sinha N, Chakrabarti R, Chattopadhyay PK. Evolutionary programming techniques for economic load dispatch. *IEEE Trans Evolut Comput* 2003;7(1):83–94.

- [23] Wood AJ, Wollenberg BF. Power generation, operation and control. NY, USA: John Wiley; 1994.
- [24] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, MA, USA: Addison Wesley; 1989.
- [25] Guo Qj, Yu Hb, Xu Ad. A hybrid PSO-GD based intelligent method for machine diagnosis. *Digit Signal Process* 2006;16(4):402–18.
- [26] Krohling RA, Hoffmann F, Coelho LS. Co-evolutionary particle swarm optimization for min-max problems using Gaussian distribution. In: *Proceedings of the congress on evolutionary computation*, Portland, Oregon, USA; 2004. p. 959–64.
- [27] Coelho LS, Krohling RA. Predictive controller tuning using modified particle swarm optimization based on Cauchy and Gaussian distribution. In: Hoffmann F, Köppen M, Klawonn F, Roy R, editors. *Soft computing: methodologies and applications, advances in soft computing series*. UK: Springer; 2005. p. 287–98.
- [28] Hénon M. A two-dimensional mapping with a strange attractor. *Commun Math Phys* 1976;50:69–77.
- [29] Stoneking D, Bilbro G, Trew R, Gilmore P, Kelley CT. Yield optimization using a GaAs process simulator coupled to a physical device model. In: *Proceedings IEEE/cornell conference on advanced concepts in high speed devices and circuits*, Ithaca, NY, USA; 1991. p. 374–83.
- [30] Bertsekas DB. On the Goldstein–Levitin–Polyak gradient projection method. *IEEE Trans Automat Contr* 1976;21(2):174–84.
- [31] Gilmore P, Kelley CT. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM J Optimiz* 1995;5(2):269–75.
- [32] Gilmore P, Kelley CT. An implicit filtering algorithm for optimization of functions with many local minima. Technical Report CRSC-TR94-23, USA: North Carolina State University, Center for Research in Scientific Computation; 1994.
- [33] Wong KP, Wong YW. Genetic and genetic/simulated-annealing approaches to economic dispatch. *IEE Proc-C* 1994;141(5):507–13.