Self-adaptive Differential Evolution Using Chaotic Local Search for Solving Power Economic Dispatch with Nonsmooth Fuel Cost Function

Leandro dos Santos Coelho and Viviana Cocco Mariani

[1] Production and Systems Engineering Graduate Program, PPGEPS
Pontifical Catholic University of Parana, PUCPR/CCET
Imaculada Conceicao, 1155, Zip code 80215-901, Curitiba, Parana, Brazil
leandro.coelho@pucpr.br
[2] Mechanical Engineering Graduate Program, PPGEM
Pontifical Catholic University of Parana, PUCPR/CCET
Imaculada Conceicao, 1155, Zip code 80215-901, Curitiba, Parana, Brazil
viviana.mariani@pucpr.br

Summary. The differential evolution (DE), proposed by Storn and Price, is a powerful population-based algorithm of evolutionary computation field designed for solving global optimization problems. The advantages of DE are its simple structure, easy use, convergence speed and robustness. However, the control parameters and learning strategies involved in DE are highly dependent on the problems under consideration. Choosing suitable parameter values requires also previous experience of the user. Despite its crucial importance, there is no consistent methodology for determining the control parameters of DE. In this chapter, different differential evolution approaches with self-adaptive mutation factor combined with a chaotic local search technique are proposed as alternative methods to solve the economic load dispatch problem of thermal units with valve-point effect. DE is used to produce good potential solutions, and the chaotic local search is used to fine-tune the DE run. DE and its variants with chaotic local search are validated for a test system consisting of 13 thermal units whose nonsmooth fuel cost function takes into account the valve-point loading effects. Numerical results indicate that performance of DE with chaotic local search presents best results when compared with previous optimization approaches in solving the load dispatch problem with the valve-point effect.

1 Introduction

The power economic dispatch problem (EDP) is one of the important problems for a power system. The objective of the EDP of electric power generation 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].

In traditional EDPs, the cost function of each generator is approximately represented by a simple quadratic function and the valve-points effects [2],[3] are ignored. These traditional EDPs are solved using mathematical programming based on several deterministic optimization techniques, such as lambda iteration, gradient method, dynamic programming, linear programming, nonlinear programming and quadratic programming [1]-[3].

However, the EDP problem with valve-point effects is represented as a nonsmooth optimization problem having complex and nonconvex features with heavy equality

U.K. Chakraborty (Ed.): Advances in Differential Evolution, SCI 143, pp. 275–286, 2008.
springerlink.com

and inequality constraints [2]. This kind of optimization problem is hard, if not impossible, to solve using traditional deterministic optimization algorithms. In other words, none of these mentioned methods may be able to provide an optimal solution, for they usually get stuck at a local optimum to the EDPs considering valve-point effects.

Recently, as an alternative to the conventional mathematical approaches, modern stochastic optimization techniques including genetic algorithms [3], evolutionary programming [4], evolution strategies [5], ant colony search algorithm [6], simulated annealing [7], and particle swarm optimization [1],[8] have been given much attention by many researchers due to their ability to find an almost global optimal solution.

In this chapter, an alternative hybrid method is proposed. The proposed hybrid method combines the differential evolution (DE) algorithm with self-adaptive mutation factor in the global search phase and a chaotic local search technique in the local search to solve the EDP associated with the valve-point effect.

DE as developed by Storn and Price [9] is one of the best evolutionary algorithms, and has proven to be a promising candidate to solve real-valued optimization problems [10]. The computational algorithm of DE is very simple and easy to implement, with only a few parameters required to be set by a user.

Chaos is a bounded unstable dynamic behavior, which exhibits sensitive dependence on initial conditions and includes infinite unstable periodic motions [11]. Optimization algorithms based on chaos theory are search methodologies that differ from all 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. The application of chaotic local search is a powerful strategy to prevent the premature convergence to local minima of DE approaches.

An EDP with 13 thermal units using nonsmooth fuel cost functions [4],[8] is employed in this chapter for demonstrate the performance of the proposed chaotic DE method. The results obtained with the DE approaches were analyzed and compared with those obtained in recent literature.

The remainder of this chapter is organized as follows. Section 2 describes the formulation of the EDP, while section 3 explains the concepts of validated optimization methods. Numerical simulation and comparisons are provided in section 5. Lastly, section 6 outlines the conclusion with a brief summary of results and future research.

2 Formulation 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 as an optimization problem (minimization) with an objective function and constraints. The equality and inequality constraints are represented by equations (1) and (2) given by:

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

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

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 load demand (in MW); P_L represents the total line losses (in MW) and P_i^{min} and P_i^{max} are, respectively, the minimum and maximum power outputs of the i-th generating unit (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.

Also in conventional methods the generating units cost functions are assumed to be convex and their incremental heat rate curves exhibit a monotonically increasing characteristics. But in reality large steam turbines have steam admission valves, which cause discontinuities in the incremental heat rate curves. Thus, the input–output characteristics of the generating units will become non-convex. Accurate modeling of the economic dispatch will be improved when the valve point loadings in the generating units are taken into account and furthermore they may generate multiple local optimum points in the cost function [1]. In this context, a more realistic cost function is obtained based on the ripple curve for more accurate modeling. This curve contains higher order nonlinearities and discontinuities due to the valve point effect, and should be refined by a sinusoidal function. Therefore, equation (4) can be modified [12], as:

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

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

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:

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

where $\tilde{F}_i$ 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, $P_L = 0$.

3 Proposed Optimization Techniques

This section describes the proposed DE approaches. First, a brief overview of DE is provided, and then the DE with self-adaptive mutation factor and chaotic local search is detailed.

3.1 Differential Evolution

Evolutionary algorithms (EAs) are general-purpose stochastic search and optimization methods that find their inspiration in the biological world. EAs differ from other optimization methods, such as Newton method, conjugate gradient, simulated annealing, by the fact that EAs maintain a population of potential (or candidate) solutions rather than a single solution to a problem.

EAs in a general sense encompass a number of related paradigms, such as genetic algorithms, evolution strategies, evolutionary programming and recently the differential evolution, all of which are based on the natural selection paradigm.

In general, all EAs work as follows: a population of individuals is randomly initialized where each individual represents a potential solution to the problem. The quality of each solution is evaluated using a fitness function. A selection process is applied during each generation of an EA in order to form a new population. The selection process is biased toward the fitter individuals in order to increase their chances of being included in the new population. Individuals are altered using unary transformation (mutation) and higher-order transformation (crossover). This procedure is repeated until convergence is reached. The best solution found is expected to be a near-optimum solution [13].

DE is a population-based stochastic function minimizer (or maximizer) relating to EAs, whose simple yet powerful and straightforward features make it very attractive for numerical optimization.

DE combines simple arithmetical operators with the classical operators of recombination, mutation and selection to evolve from a randomly generated starting population to a final solution. DE uses mutation which is based on the distribution of solutions in the current population. In this way, search directions and possible step sizes depend on the location of the individuals selected to calculate the mutation values [14]. It evolutes generation by generation until the termination conditions have been met.

The different variants of DE are classified using the following notation: DE/α/β/δ, where α indicates the method for selecting the parent chromosome that will form the base of the mutated vector, β indicates the number of difference vectors used to perturb the base chromosome, and δ indicates the recombination mechanism used to create the offspring population. The *bin* acronym indicates that the recombination is controlled by a series of independent binomial experiments.

The fundamental idea behind DE is a scheme whereby it generates the trial parameter vectors. In each step, the DE mutates vectors by adding weighted, random vector differentials to them. If the cost of the trial vector is better than that of the target, the target vector is replaced by the trial vector in the next generation. The variant implemented here was DE/*rand*/1/*bin*, which involved the following steps and procedures:

Step 1: *Initialization of the parameter setup*: The user must choose the key parameters that control DE, i.e., population size, boundary constraints of optimization variables, mutation factor (f_m), crossover rate (CR), and the stopping criterion (t_{max}).

Step 2: *Initialize the initial population of individuals*: Initialize the generation's counter t= 0 and also initialize a population of individuals (solution vectors) $x(t)$ with random values generated according to a uniform probability distribution in the n-dimensional problem space.

Step 3: *Evaluate the objective function value*: For each individual, evaluate its objective function (fitness) value.

Step 4: *Mutation operation (or differential operation)*: Mutate individuals according to the following equation:

$$z_i(t+1) = x_{i,r_1}(t) + f_m \cdot [x_{i,r_2}(t) - x_{i,r_3}(t)] \tag{8}$$

where i =1,2,...,N is the individual's index of population; t is the generation counter (time or iteration); $f_m > 0$ is a real parameter, called *mutation factor*, which controls the amplification of the difference between two individuals and it is usually taken form the range [0.1, 1]; $x_i(t) = \left[x_{i_1}(t), x_{i_2}(t), ..., x_{i_n}(t)\right]^{\mathrm{T}}$ stands for the i-th individual of population of N real-valued n-dimensional vectors; $z_i(t) = \left[z_{i_1}(t), z_{i_2}(t), ..., z_{i_n}(t)\right]^{\mathrm{T}}$ stands for the i-th individual of a *mutant vector*; r_1, r_2 and r_3 are mutually different integers and also different from the running index, i, randomly selected with uniform distribution from the set $\{1, 2, \cdots, i-1, i+1, \cdots, N\}$.

Step 5: *Crossover (recombination) operation*: Following the mutation operation, crossover is applied in the population. For each mutant vector, $z_i(t+1)$, an index $rnbr(i) \in \{1, 2, \cdots, n\}$ is randomly chosen using a uniform distribution, and a *trial vector*, $u_i(t+1) = \left[u_{i_1}(t+1), u_{i_2}(t+1), ..., u_{i_n}(t+1)\right]^T$, is generated via

$$u_{i_j}(t+1) = \begin{cases} z_{i_j}(t+1), & \text{if } randb(j) \le CR \text{ or } j = rnbr(i), \\ x_{i_j}(t), & \text{if } randb(j) > CR \text{ or } (j \ne rnbr(i). \end{cases} \tag{9}$$

where j=1,2,..., n is the parameter index; $x_{ij}(t)$ stands for the i-th individual of *j-th* real-valued vector; $z_{ij}(t)$ stands for the i-th individual of *j-th* real-valued vector of a *mutant vector*; $u_{ij}(t)$ stands for the i-th individual of *j-th* real-valued vector after crossover operation; *randb(j)* is the j-th evaluation of a uniform random number generation with [0, 1]; CR is a *crossover rate* in the range [0, 1].

To decide whether or not the vector $u_i(t + 1)$ should be a member of the population comprising the next generation, it is compared to the corresponding vector $x_i(t)$. Thus, if f denotes the objective function under minimization, then

$$x_i(t+1) = \begin{cases} u_i(t+1), & \text{if } f(u(t+1)) < f(x_i(t)), \\ x_i(t), & \text{otherwise} \end{cases} \tag{10}$$

Step 6: *Update the generation's counter*: $t = t + 1$;

Step 7: *Verification of the stopping criterion*: Loop to **Step 2** until a stopping criterion is met, usually a maximum number of iterations (generations), t_{max}.

3.2 Self-adaptive Differential Evolution Approaches

The parameters *CR* and *fm* of DE are generally the key factors affecting the DE's convergence [13],[15],[16]. In this chapter, we use a self-adaptive control mechanism to change the mutation factor *fm* during the run. The control parameters *M* and *CR* are not changed during the run. In this context, the DE/*rand*/1/*bin* algorithm based on self-adaptive mutation factor is proposed in this work. Several DE-variants are used in this work for comparison purposes:

- DE(1): classical DE using a constant mutation factor of $f_m = 0.50$;
- DE(2): classical DE using a constant mutation factor of $f_m = 0.75$;
- DE(3): classical DE using a constant mutation factor of $f_m = 1.00$;
- ADE(1): adaptive DE using a linear increase of f_m with initial and final values of 0.5 and 1.0, respectively;
- ADE(2): adaptive DE using a linear reduction of f_m with initial and final values of 1.0 and 0.5, respectively;
- ADE(3): adaptive DE using a mutation factor f_m generated by random number with uniform distribution in the range [0.5, 1];
- ADE(4): adaptive DE using a mutation factor f_m generated by random number with Gaussian distribution and normalized in the range [0.5, 1].

3.3 Chaotic Local Search

Chaos theory is recognized as very useful in many optimization applications. An essential feature of chaotic systems is that small changes in the parameters or the starting values for the data lead to vastly different future behaviors, such as stable fixed points, periodic oscillations, bifurcations, and ergodicity.

This sensitive dependence on initial conditions of chaotic systems is generally exhibited by systems containing multiple elements with nonlinear interactions, particularly when the system is forced and dissipative. Sensitive dependence on initial conditions is not only observed in complex systems, but even in the simplest logistic equation [17].

The application of chaotic sequences in DE approaches can be a good alternative to maintain the search diversity in an optimization procedure. Due to the non-repetition of chaos, it can carry out overall searches at higher speeds than stochastic ergodic searches that depend on probabilities [18]-[20].

Different types of equations of chaotic systems have been considered in the literature for applications in optimization methods. The logistic equation and other equations, such as sinusoidal iterator, Chua's oscillator, Lorenz system, Ikeda map, and others, have been adopted instead of generation of random numbers using a uniform distribution and very interesting results have emerged [18]-[20]. The design of approaches to improve the convergence of chaotic optimization is a challenging issue. A chaotic local search approach is proposed here based on Lozi map [19].

The Lozi's piecewise liner model is a simplification of the Hénon map [21] and it admits strange attractors. The Lozi map is given by

$$y_1(k)=1-a\cdot\left|y_1(k-1)\right|+y(k-1) \tag{11}$$

$$y(k)=b\cdot y_1(k-1) \tag{12}$$

where k is the iteration number. In this work, the values of y are normalized in the range [0,1] to each i-th decision variable. This transformation is given by

$$w_i(k)=\frac{y(k)-\alpha}{\beta-\alpha} \tag{13}$$

where $y\in[-0.6418,0.6716]$ and (α,β) = (-0.6418,0.6716). The parameters used in this work are a=1.7 and b=0.5, as these values have been suggested by [19].

The chaotic search procedure based on the Lozi map can be illustrated as follows:

Notation:

$X=[x_1,x_2,...,x_n]$: solution vector consisting of n variables x_i , i = 1,…,n bounded by lower (L_i) and upper limits (U_i).

Input:

M_L: maximum number of iterations of chaotic *L*ocal search;

λ: step size in chaotic local search.

Output:

X_i*: best solution of *j-th* variable from current run of chaotic search;

*f**: best objective function (minimization problem).

Chaotic optimization algorithm:

Step 1: *Initialization of variables*: Set k = 0, where k represents the iteration number. Set the initial conditions $y_1(0)$,$y(0)$, a=1.7 and b=0.5 of Lozi map. Set the initial best objective function f^*. In this work, the best objective function is the best individual of differential evolution in current generation t;

Step 2: *Exploitation phase of chaotic search*:

Begin
While $k \le M_L$ do
  For i =1 to n
    If $r < 0.5$ then
    (where r is a uniformly distributed random variable in [0, 1])
    $$x_i(k)=X_i^*+\lambda\cdot w_i(k)\cdot\left|U_i-X_i^*\right|$$
    Else If
    $$x_i(k)=X_i^*-\lambda\cdot w_i(k)\cdot\left|X_i^*-L_i\right|$$
    End If
  End
  If $f(X(k))<f^*$ then
    $X^*=X(k)$
    $f^*=f(X(k))$

End If
$k = k + 1;$
End
End

During the exploitation phase of chaotic search, the step size λ is an important parameter for the convergence behavior of the optimization method, which adjusts small ranges around X^*. A suitable value for the step size usually provides a balance between global and local search abilities and consequently a reduction on the number of iterations required to locate the optimum solution. In this work, the step size λ = 0.0001 is adopted in chaotic local search (CLS).

3.4 Differential Evolution with Chaotic Local Search

The approaches configuration composite by DE hybridized with stochastic techniques is a promising alternative in optimization and must be evaluated. DE and the proposed chaotic local method have supplementary potentialities. In this work, the following way of hybridizing of DE combined with CLS was tested: after having solved the EDP use the best solution from DE as a starting point and solve the EDP using CLS method.

4 Simulation Results

In this section, we judge the performance of the DE and DE-CLS algorithms using a case study of power economic dispatch using 13 thermal units.

This case study consisted of 13 thermal units of generation with the effects of valve-point loading, as given in Table 1. The data shown in Table 1 is also available in [4] and [22]. In this case, the load demand expected to be determined was $P_D = 1800$ MW. This EDP has many local minima, and the global minimum is difficult to determine.

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 2 GB 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.

A key factor in the application of DE approaches is how the algorithm handles the constraints relating to the problem. In this work, a penalty-based method proposed in [23] was used for the equality constraints.

The population size N was 20 and the stopping criterion t_{max} was 800 generations (16000 evaluations of the objective function) for classical DE.

In the DE-CLS, the population size of DE was 12 and the stopping criterion t_{max} was 500 generations. CLS procedure is adopted using 12 cost function evaluations (M_L = 12) in each generation of DE. In this case, the DE-CLS routine is adopted using 16000 cost function evaluations in each run. The crossover rate of CR = 0.8 was adopted for both the classical DE and DE-CLS approaches.

The results obtained for this case study are given in Table 2, which shows that the DE(3)-CLS succeeded in finding the best solution for the tested methods. The best result obtained for solution vector P_i, i=1,..,13 with DE(3)-CLS is the minimum cost of 17963.9571 which is given in Table 3. However, the ADE(1)-CLS approach shows a performance which is clearly better than that of DE(3)-CLS in terms of mean cost.

It also observed that the classical DE approaches outperformed the other tested DE-CLS methods in terms of solution time.

Table 4 compares the results obtained in this chapter with those of other studies reported in the literature. Note that in the case studied here, the best result reported using DE(3)-CLS is comparatively lower than recent studies presented in the literature.

Table 2. Convergence results (50 runs) of DE and DE-CLS approaches

Optimization Method	Mean Time (s)	Minimum Cost ($/h)	Mean Cost ($/h)	Maximum Cost ($/h)
DE(1)	1.78	18095.7270	18323.9653	**18637.0927**
DE(2)	1.77	18091.1464	18315.6026	18682.1625
DE(3)	1.82	18377.7128	18752.4246	19116.6163
ADE(1)	1.78	18052.7891	18294.6310	18645.2262
ADE(2)	1.77	18069.1528	18419.8325	18903.2219
ADE(3)	1.79	18097.9214	18302.1210	18646.4057
ADE(4)	1.79	18070.2032	18337.7369	18782.8841
DE(1)-CLS	5.39	18085.5078	18427.0199	18815.4248
DE(2)-CLS	5.38	18089.7461	18327.8504	18623.3178
DE(3)-CLS	5.37	**17963.9571**	18431.1479	18892.7540
ADE(1)-CLS	5.37	18001.7035	**18274.9005**	18524.9235
ADE(2)-CLS	5.37	18093.4723	18424.7626	18782.6906
ADE(3)-CLS	5.36	18101.2664	17320.5504	17683.0652
ADE(4)-CLS	5.36	18057.9074	18371.7782	18786.6667

Table 3. Best result (50 runs) obtained for the case study using DE(3)-CLS

Power	Generation (MW)	Power	Generation (MW)
P_1	628.3180	P_8	60.0000
P_2	149.1094	P_9	109.8664
P_3	223.3226	P_{10}	40.0000
P_4	109.8650	P_{11}	40.0000
P_5	109.8618	P_{12}	55.0000
P_6	109.8656	P_{13}	55.0000
P_7	109.7912	$\sum_{i=1}^{13} P_i$	1800.0000

Table 4. Comparison of best results for fuel costs presented in the literature

Optimization Technique	Best Objective Function
Evolutionary programming [4]	17994.07
Particle swarm optimization [1]	18030.72
Hybrid evolutionary programming with SQP [1]	17991.03
Hybrid particle swarm with SQP [1]	17969.93
Genetic algorithms [24]	17975.3437
Improved genetic algorithm with multiplier updating [24]	17963.9848
Best result of this chapter using DE(3)-CLS	**17963.9571**

5 Conclusion and Future Research

In this chapter, DE and DE-CLS methods have been successfully introduced to solve a case study of EDP considering 13 thermal units with valve-point effect. In this case study, DE, DE-CLS and ADE-CLS can provide accurate dispatch solutions in reasonable time.

In relation to procedure of solution of the economic dispatch problem of electric energy with effect of valve point, the results with the DE(3)-CLS for optimization of the equations (1) and (2) were best that the results presented in [1], [4] and [24].

Future research will investigate theoretically the effect of chaos incorporation into DE further and apply the DE-CLS methods for solving the multiobjective economic dispatch problems in power systems.

Acknowledgments

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, T.A.A., Jeyakumar, A.E.: Hybrid PSO-SQP for economic dispatch with valve-point effect. Electric Power Systems Research 71(1), 51–59 (2004)
2. Al-Othman, A.K., El-Naggar, K.M.: Application of pattern search method to power security constrained economic dispatch with non-smooth cost function. Electric Power Systems Research 78(4), 667–675 (2008)
3. Walters, D.C., Sheble, G.B.: Genetic algorithm solution of economic dispatch with valve point loading. IEEE Transactions on Power Systems 8(3), 1325–1332 (1993)
4. Sinha, N., Chakrabarti, R., Chattopadhyay, P.K.: Evolutionary programming techniques for economic load dispatch. IEEE Transactions on Evolutionary Computation 7(1), 83–94 (2003)
5. Gomes, J.R., Saavedra, O.R.: A Cauchy-based evolution strategy for solving the reactive power dispatch problem. Electrical Power and Energy Systems 24(4), 277–283 (2002)
6. Sum-im, T.: Economic dispatch by ant colony search algorithm. In: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, Singapore, pp. 416–421 (2004)
7. Wong, K.P., Wong, Y.W.: Thermal generator scheduling using hybrid genetic/simulated-annealing approach. IEE Proc.-Generation, Transmission and Distribution 142(4), 372–380 (1995)
8. Park, J.-B., Lee, K.-S., Shin, J.-R., Lee, K.Y.: A particle swarm optimization for economic dispatch with nonsmooth cost function. IEEE Transactions on Power Systems 20(1), 34–42 (2005)
9. Storn, R., Price, K.: Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, USA (1995)
10. Storn, R.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
11. Ho, S.J., Shu, L.S., Ho, S.Y.: Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA. IEEE Transactions on Fuzzy Systems 14(3), 421–434 (2006)
12. Wood, A.J., Wollenberg, B.F.: Power generation, operation and control. John Wiley & Sons, New York (1994)
13. Anzi, F.S., Allahverdi, A.: A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. European Journal of Operation Research (accepted for future publication, 2007)
14. Montes, E.M., Reyes, J.V., Coello, C.A.C.: A comparative study of differential evolution variants for global optimization. In: Proceedings of Genetic and Evolutionary Computation Conference, Seattle, Washington, USA (2006)
15. Liu, J., Lampinen, J.: On setting the control parameter of the differential evolution method. In: Proceeding of 8th International Conference on Soft Computing (MENDEL 2002), Brno, Czech Republic, pp. 11–18 (2002)
16. Brest, J., Saso, G., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation (accepted for future publication, 2007)
17. Yan, X.F., Chen, D.Z., Hu, S.X.: Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. Computers and Chemical Engineering 27(10), 1393–1404 (2003)

18. Pan, H., Wang, L., Liu, B.: Chaotic annealing with hypothesis test for function optimization in noisy environments. Chaos, Solitons & Fractals (accepted for future publication, 2007)
19. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.G.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(3), 289–304 (2003)
20. Li, L., Yang, Y., Peng, H., Wang, X.: Parameters identification of chaotic systems via chaotic ant swam. Chaos, Solitons & Fractals 28(5), 1204–1211 (2006)
21. Hénon, M.: A two dimensional mapping with a strange attractor. Communications in Mathematical Physics 50, 69–77 (1976)
22. Wong, K.P., Wong, Y.W.: Genetic and genetic/simulated-annealing approaches to economic dispatch. IEE Proc. Control, Generation, Transmission and Distribution 141(5), 507–513 (1994)
23. Coelho, L.S., Mariani, V.C.: Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Transactions on Power Systems 21(2), 989–996 (2006)
24. Chiang, C.L.: Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. IEEE Transactions on Power Systems 20(4), 1690–1699 (2005)