Volume 7, Issue 5, May 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Modified Differential Evolution Algorithm for Solving Non-Convex Dynamic Economic Dispatch Problems

Hardiansyah[*1], Fuazen[2]
[1] Department of Electrical Engineering, University of Tanjungpura, Indonesia
[2] Department of Mechanical Engineering, University of Muhammadiyah Pontianak, Indonesia

DOI: 10.23956/ijarcsse/SV7I5/0285

Abstract— In this paper, a modified differential evolution (MDE) algorithm is presented to solve non-convex dynamic economic dispatch (DED) problems considering valve-point effects, the ramp rate limits and transmission losses. The practical DED problems have non-smooth cost function with equality and inequality constraints, which make the problem of finding the global optimum difficult when using any mathematical approaches. The proposed MDE algorithm is inspired by three ideas: (1) use of opposition based learning to generate the initial population, (2) use of tournament best process to generate mutant vector to explore the region around the tournament best individual, (3) use of a single set population in contrast to the two set population as in basic DE. The feasibility of the proposed method is validated on 5 and 10 units test system for a 24 h time interval. The results are compared with the results reported in the literature.

Keywords— Modified differential evolution, dynamic economic dispatch, non-smooth cost functions, ramp rate limits, valve-point effects

I. INTRODUCTION

The electrical power systems are interconnected in order to obtain the benefits of minimum generation costs, maximum reliability and best operational conditions, such as sharing of power reserve, improving the stability and operating on emergency situations. Thus, the optimization problem of the economic dispatch of electrical power system is relevant to accomplish requirements of quality and efficiency in power generation. Dynamic economic dispatch (DED) is one of important problems in power system operation and control, which is used to determine the optimal schedule of generating outputs on-line so as to meet the load demand at the minimum operating cost under various system and operating constraints over the entire dispatch periods. DED is an extension of the conventional economic dispatch (ED) problem that takes into consideration the limits on the ramp rate of generating units to maintain the life of generation equipment [1, 2].

Since the DED problem was introduced, several optimization techniques and procedures have been used for solving the DED problem with complex objective functions or constraints. There were a number of classical methods that have been applied to solve this problem such as gradient projection method, Lagrange relaxation, and linear programming [3-5]. Most of these methods are not applicable for non-smooth or non-convex cost functions. To overcome this problem, many stochastic optimization methods have been employed to solve the DED problem, such as genetic algorithm (GA) [6], simulated annealing (SA) [7], differential evolution (DE) [8, 9], particle swarm optimization (PSO) [10], hybrid EP and SQP [11], deterministically guided PSO [12], hybrid PSO and SQP [13], and imperialist competitive algorithm (ICA) [14]. Many of these techniques have proven their effectiveness in solving the DED problem without any or fewer restrictions on the shape of the cost function curves.

Differential evolution (DE) algorithm introduced by Storn and Price in 1995, belongs to the group of evolutionary algorithms which operate in continuous search spaces [15, 16]. This algorithm has high efficiency for solving continuous nonlinear optimization problems and multimodal environments. The advantages of the DE are simple structure, a few control parameters and high reliable convergences. The DE is one type of modern optimization techniques, which based on a population searching mechanism like as GA and PSO.

In this paper, a novel approach is proposed to solve the DED problem with valve-point effects using a modified differential evolution (MDE) algorithm. The proposed method considers the nonlinear characteristics of a generator such as valve-point effects, the ramp rate limits and transmission losses. Feasibility of the proposed MDE method has been demonstrated on two different test systems, i.e. 5 and 10 generating unit systems. Results obtained show that the proposed approach can obtain more optimum solutions and the results are compared with other methods reported in recent literature in order to demonstrate its performance.

II. DED PROBLEM FORMULATION

The objective of DED problem is to find the optimal schedule of output powers of online generating units with predicted power demands over a certain period of time to meet the power demand at minimum operating cost.

The fuel cost function of the generating unit is expressed as a quadratic function of real power generation. The objective function of the DED problem is

$$\min F_T = \sum_{t=1}^{T}\sum_{i=1}^{N} F_{i,t}(P_{i,t}) = \sum_{t=1}^{T}\sum_{i=1}^{N}\left(a_i P_{i,t}^2 + b_i P_{i,t} + c_i\right) \quad (1)$$
$$\text{for } i = 1,2,\cdots,N;\ t = 1,2,\ldots,T$$

where $F_{i,t}$ is the fuel cost of unit i at time interval t in \$/hr, a_i, b_i, and c_i are the cost coefficients of generating unit i, $P_{i,t}$ is the real power output of generating unit i at time period t in MW, and N is the number of generators. T is the total number of hours in the operating horizon.

The valve-point effects are taken into consideration in the DED problem by superimposing the basic quadratic fuel-cost characteristics with the rectified sinusoid component as follows:

$$\min F_T = \sum_{t=1}^{T}\sum_{i=1}^{N} F_{i,t}(P_{i,t}) = \sum_{t=1}^{T}\sum_{i=1}^{N}\left(a_i P_{i,t}^2 + b_i P_{i,t} + c_i + \left|e_i \times \sin\left(f_i \times \left(P_{i,\min} - P_{i,t}\right)\right)\right|\right) \quad (2)$$

where F_T is total fuel cost of generation in (\$/hr) including valve point loading, e_i, f_i are fuel cost coefficients of unit i reflecting valve-point effects.

The fuel cost is minimized subjected to the following constraints:

2.1 Active Power Balance Equation

For power balance, an equality constraint should be satisfied. The total generated power should be the same as total load demand plus the total line loss.

$$\sum_{i=1}^{N} P_{i,t} = P_{D,t} + P_{L,t} \quad (3)$$

where $P_{D,t}$ and $P_{L,t}$ are the load demand and transmission loss in MW at time interval t, respectively.

The transmission loss $P_{L,t}$ can be expressed by using ***B*** matrix technique and is defined by (4) as,

$$P_{L,t} = \sum_{i=1}^{n}\sum_{j=1}^{n} P_{i,t} B_{ij} P_{j,t} + \sum_{i=1}^{n} B_{0i} P_{i,t} + B_{00} \quad (4)$$

where B_{ij}, B_{0i}, and B_{00} are coefficient of transmission loss.

2.2 Minimum and Maximum Power Limits

Generation output of each generator should lie between minimum and maximum limits. The corresponding inequality constraint for each generator is

$$P_{i,\min} \le P_{i,t} \le P_{i,\max} \quad (5)$$

where $P_{i,min}$ and $P_{i,max}$ are the minimum and maximum real power output of unit i in MW, respectively.

2.3 Ramp Rate Limits

The actual operating ranges of all on-line units are restricted by their corresponding ramp rate limits. The ramp-up and ramp-down constraints can be written as (6) and (7), respectively.

$$P_{i,t} - P_{i,t-1} \le UR_i \quad (6)$$

$$P_{i,t-1} - P_{i,t} \le DR_i \quad (7)$$

where $P_{i,t}$ and $P_{i,t-1}$ are the present and previous power outputs, respectively. UR_i and DR_i are the ramp-up and ramp-down limits of unit i (in units of MW/time period). To consider the ramp rate limits and power output limits constraints at the same time, therefore, eqs. (5), (6) and (7) can be rewritten as follows:

$$\max\{P_{i,\min}, P_{i,t-1} - DR_i\} \le P_{i,t} \le \min\{P_{i,\max}, P_{i,t-1} + UR_i\} \quad (8)$$

III. DIFFERENTIAL EVOLUTION (DE) ALGORITHM

Differential evolution (DE) developed by Storn and Price [15] is a population based evolutionary computation technique, capable of handling non-differentiable, non-linear and multi-modal objective functions. Due to its simple but powerful and straightforward features, it is very attractive for resolving the non-convex global optimization problems. In DE, the fitness of an offspring competes one-to-one with that of the corresponding parent. This one-to-one competition will give rise to a faster convergence rate than other evolutionary algorithms (EAs). In addition, only a few control parameters are required in comparison with other computing heuristic optimization methods [16]. The basic algorithm of DE typically consists of four phases: 1) initialization, 2) mutation, 3) crossover, and 4) selection phases. The mutation and crossover are used to generate new individuals, and the selection then determines that the individuals will survive into the next generation. The performance of DE algorithm usually depends on three parameters, i.e., population size *NP*, mutation factor *MF*, and crossover rate *CR*.

A brief description of different steps of DE algorithm is given below:

3.1. Initialization

The population is initialized by randomly generating individuals within the boundary constraints

$$X_{ij}^{0} = X_j^{\min} + rand * \left(X_j^{\max} - X_j^{\min}\right)$$
$$i = 1,2,\cdots,N_p\ ;\ j = 1,2,\cdots,D \quad (9)$$

where X_{ij}^{0} is the initialized *j*th decision variable of *i*th population set; 'rand' function generates random values uniformly in the interval [0, 1]; *Np* is the size of the population; *D* is the number of decision variables. The fitness function is evaluated for each individual. $X_j^{\min}$ and $X_j^{\max}$ are the lower and upper bound of the *j*th decision variable, respectively.

3.2. Mutation

As a step of generating offspring, the operations of 'mutation' are applied. 'Mutation' occupies quite an important role in the reproduction cycle. The mutation operation creates mutant vectors $X_i^{'k}$ by perturbing a randomly selected vector X_a^k with the difference of two other randomly selected vectors X_b^k and X_c^k at *k*th iteration as per following equation.

$$X_i^{'k} = X_a^k + F * \left(X_b^k - X_c^k\right) \quad (10)$$
$$i = 1,2,\cdots,N_p$$

where $X_i^{'k}$ is a newly generated *i*th population set after performing mutation operation at *k*th iteration; X_a^k, X_b^k and X_c^k are randomly chosen vectors at *k*th iteration $\in (=1,2,\cdots,N_p)$ and $a \neq b \neq c \neq i$. X_a^k, X_b^k and X_c^k are selected for each new parent vector.

$F \in [0, 2]$ is known as 'scaling factor' used to control the amount of perturbation in the mutation process and improve convergence. Many schemes of creation of a candidate are possible. Here strategy 1 has been mentioned in the algorithm.

3.3. Crossover

Crossover represents a typical case of a 'genes' exchange. The parent vector is mixed with the mutated vector to create a trial vector, according to the following equation:

$$X_i^{''k} = \begin{cases} X_{ij}^{'k} \text{ if } rand\ j < Cr \text{ or } j = q \\ X_i^k \quad otherwise \end{cases} \quad (11)$$

where *i*=1,2,…, *Np* ; *j*=1, …, *D*. $X_{ij}^k, X_{ij}^{'k}$, and $X_{ij}^{''k}$ are the *j*th individual of *i*th target vector, mutant vector, and trial vector at *k*th iteration, respectively. q is a randomly chosen index $\in$ (*j* = 1, 2, …, *D*) that guarantees that the trial vector gets at least one parameter from the mutant vector even if *Cr* = 0. *Cr* = [0, 1] is the 'Crossover constant' that controls the diversity of the population and aids the algorithm to escape from local optima.

3.4. Selection

Selection procedure is used among the set of trial vector and the updated target vector to choose the best. Each solution in the population has the same chance of being selected as parents. Selection is realized by comparing the objective function values of target vector and trial vector. For minimization problem, if the trial vector has better value of the objective function, then it replaces the updated one as:

$$X_i^{k+1} = \begin{cases} X_i^{''k} \text{ if } X_i^{''k} \leq f(X_i^k) \\ X_i^k \quad otherwise \end{cases} \quad (12)$$

where X_i^{k+1} is the *i*th population set obtained after selection operation at the end of *k*th iteration, to be used as parent population set (in *i*th row of population matrix) in next iteration (*k* +1 th).

IV. MODIFIED DIFFERENTIAL EVOLUTION

In this section we describe the proposed MDE, which uses the concepts of opposition based learning, random localization and one population set. The basic operators of MDE are same as basic DE but still it is different from it three points [17]:

1. MDE differs from basic DE in the initialization phase where MDE utilizes opposition based learning method while DE uses uniform random numbers for initialization of population.
2. In mutation step MDE uses best individual of three points as mutant individual while in DE it is random (there is an equal chance of all these three for being selected as mutant individual).
3. MDE maintain one population set while DE maintains two population sets, one current population and second advanced population (for next generation). The population is updated as the better individual is found. Also the newly found individual can take part in generation of new individual in current generation.

<u>MDE Procedure:</u>

Initialization: Randomly construct a population *P* of *NP* individual, dimension of each vector being *n,* using the following rule,

$$X_{ij} = X_j^{\min} + rand(0,1) * \left(X_j^{\max} - X_j^{\min}\right) \quad (13)$$

where $X_j^{\min}$ and $X_j^{\max}$ are lower and upper bound for *j*th component respectively and *rand* (0, 1) is a uniform random number between 0 and 1.

Construct another population *OP* of *NP* individual using the following rule:

$$y_{ij} = X_j^{\min} + X_j^{\max} + P_{ij} \tag{14}$$

where P_{ij} are the points of population *P*.
Construct initial population *S* taking *NP* best individuals from union of these two populations.
Mutation: Select randomly three distinct individuals X_{tb}, X_{r2}, and X_{r3} from population *S* and perform mutation using formula,

$$V_i = X_{tb} + F * (X_{r2} - X_{r3}) \tag{15}$$

where individual X_{tb} is best of these three individuals and X_{r2}, X_{r3} are the remaining two.
Crossover: Perform crossover according to equation (11).
Selection: Calculate the objective function value at new generated individual. If it is better than target individual then replace target individual by this new individual in current population.

V. SIMULATION RESULTS AND DISCUSSIONS

The DED problem was solved using the MDE algorithm and its performance is compared with other methods reported in recent literature. The proposed technique has been applied to 5 and 10 unit test systems. The algorithm was implemented in MATLAB 7.1 on a Pentium IV personal Computer with 3.6 GHz speed processor and 2 GB RAM. For all cases, the dispatch horizon is selected as one day with 24 dispatch periods of each one hour.

1) Case Study 1

The first test system is a 5-unit test system. The technical data of the units are taken from [9]. In this test system, valve-point effect, the ramp rate limits, and transmission losses are considered. The load demand for each time interval over the scheduling period is given in Table 1. The optimal dispatch of real power for the given scheduling horizon using MDE algorithm is given in Table 2. The best solution obtained through the proposed method is compared to those reported in the recent literature. The best total production cost obtained using proposed method is $ 41479.3026 and the computation time taken by the algorithm is 3.8565s. Table 3 shows the comparison results for different methods.

Table 1 Load demand for 24 hours (5-unit system)

Time (h)	Load (MW)	Time (h)	Load (MW)	Time (h)	Load (MW)	Time (h)	Load (MW)
1	410	7	626	13	704	19	654
2	435	8	654	14	690	20	704
3	475	9	690	15	654	21	680
4	530	10	704	16	580	22	605
5	558	11	720	17	558	23	527
6	608	12	740	18	608	24	463

Table 2 Best scheduling of 5-unit system using MDE method

Hour	P1 (MW)	P2 (MW)	P3 (MW)	P4 (MW)	P5 (MW)	Cost ($)	Ploss (MW)
1	10.0011	27.6266	37.9710	126.7157	211.5552	1221.3229	3.8696
2	10.0009	56.8832	60.3402	131.9425	179.9644	1354.2381	4.1313
3	37.8107	111.2250	94.5166	129.6465	106.5934	1363.5941	4.7922
4	31.0429	102.8769	103.6222	153.5984	144.7989	1577.9430	5.9392
5	33.1080	103.7348	175.0000	132.6172	120.0343	1563.7362	6.4943
6	26.9756	99.8617	121.0521	184.3412	183.5955	1769.3794	7.8261
7	50.7991	125.0000	147.1093	183.5160	127.8391	1772.9719	8.2636
8	10.0326	74.4407	122.3529	208.8807	247.5025	1790.1798	9.2095
9	31.0078	109.9202	145.2413	212.2439	201.6638	1971.9547	10.0769
10	51.2713	125.0000	168.4742	213.8382	155.8498	1979.9012	10.4334
11	57.4586	125.0000	175.0000	223.3787	150.0821	1963.0444	10.9194
12	10.0120	68.1535	139.6326	241.5511	292.5185	2020.1013	11.8678
13	51.2709	125.0000	168.4699	213.8377	155.8550	1979,9012	10.4335
14	31.0035	109.9261	145.2668	212.2273	201.6531	1971.9547	10.0768
15	10.1022	73.2408	122.3626	211.5868	245.9186	1790.1866	9.2110
16	10.0030	48.2833	90.5640	185.5993	252.9331	1637.9361	7.3827
17	39.6510	118.1686	117.5559	158.2154	130.9938	1557.9161	6.5847
18	26.9803	99.8612	121.0480	184.3423	183.5943	1769.3794	7.8261
19	10.0212	74.0459	122.6711	209.4493	247.0204	1790.1778	9.2079

20	51.1447	125.0000	172.6456	211.6031	154.0286	1979.9304	10.4220
21	13.0879	81.1286	131.4661	217.5445	246.6912	1938.8376	9.9183
22	21.1439	90.4718	116.4251	186.2967	198.4423	1762.1006	7.7798
23	25.4582	93.8375	98.8923	155.5561	159.1399	1573.9876	5.8839
24	44.4158	121.5012	97.0185	121.1429	83.5151	1378.6276	4.5935
Total						**41479.3026**	**193.1438**

Table 3 Comparison of results for 5-unit systems

Method	Production cost ($)	Computing time (s)
SA [7]	47356	351.98
DE [8]	43213	376
IDE [9]	45800	197
PSO [10]	50124	258.00
Proposed	41479.3026	3.8565

2) Case Study 2

The second test system is a 10-unit test system. In this case, generator capacity limits, ramp rate constraints and valve-point effects are considered. The transmission losses are ignored in this case for sake of comparison. The data for this system can be found from [9]. The load demand for each time interval over the scheduling period is given in Table 4. The optimal dispatch of real power for the given scheduling horizon using MDE algorithm is given in Table 5. The best solution obtained through the proposed method is compared to those reported in the recent literature are shown in Table 6. The best total production cost obtained using proposed method is $ 1001746.4962 and the computation time taken by the algorithm is 4.2798s. It clear from the table that the proposed method produces much better results compared to recently reported different method for solving DED problem.

Table 4 Load demand for 24 hours (10-unit system)

Time (h)	Load (MW)	Time (h)	Load (MW)	Time (h)	Load (MW)	Time (h)	Load (MW)
1	1036	7	1702	13	2072	19	1776
2	1110	8	1776	14	1924	20	2072
3	1258	9	1924	15	1776	21	1924
4	1406	10	2072	16	1554	22	1628
5	1480	11	2146	17	1480	23	1332
6	1628	12	2220	18	1628	24	1184

Table 5 Best scheduling of 10-unit system using MDE method

Hour	P1 (MW)	P2 (MW)	P3 (MW)	P4 (MW)	P5 (MW)	P6 (MW)	P7 (MW)	P8 (MW)	P9 (MW)	P10 (MW)
1	151.3266	159.0253	173.1142	60.0000	73.0000	160.0000	130.0000	48.0238	26.5101	55.0000
2	150.1436	135.0000	228.7354	60.0000	124.1211	160.0000	130.0000	47.0000	20.0000	55.0000
3	155.4697	216.5876	340.0000	60.0000	73.0000	160.0000	130.0000	47.0000	20.9427	55.0000
4	150.0271	332.4620	340.0000	60.0000	111.5108	160.0000	130.0000	47.0000	20.0000	55.0000
5	150.0002	388.7997	340.0000	60.0000	129.2001	160.0000	130.0000	47.0000	20.0000	55.0000
6	200.8316	460.0000	340.0000	60.0000	155.1684	160.0000	130.0000	47.0000	20.0000	55.0000
7	206.4189	415.3583	340.0000	60.0000	243.0000	160.0000	130.0000	68.5009	23.7218	55.0000
8	339.0505	460.0000	340.0000	60.0000	164.9495	160.0000	130.0000	59.0390	20.0000	55.0000
9	369.0081	460.0000	340.0000	60.0000	243.0000	160.0000	130.0000	86.9919	20.0000	55.0000
10	469.9000	460.0000	340.0000	117.7022	243.0000	160.0000	130.0000	75.4685	20.9292	55.0000
11	469.999	460.000	340.000	216.838	243.000	160.000	130.000	47.0000	24.1622	55.000

	6	0	0	2	0	0	0			0
12	469.9977	460.0000	340.0000	212.5109	243.0000	160.0000	130.0000	120.0000	29.4914	55.0000
13	469.9996	460.0000	340.0000	117.0722	243.0000	160.0000	130.0000	76.9283	20.0000	55.0000
14	370.6530	460.0000	340.0000	60.0000	243.0000	160.0000	130.0000	81.1320	24.2150	55.0000
15	339.0505	460.0000	340.0000	60.0000	164.9495	160.0000	130.0000	47.0000	20.0000	55.0000
16	150.0811	445.0353	340.0000	60.0000	146.8836	160.0000	130.0000	47.0000	20.0000	55.0000
17	150.0033	273.8833	340.0000	60.0000	243.0000	160.0000	130.0000	47.0000	221.1134	55.0000
18	200.8316	460.0000	340.0000	60.0000	155.1684	160.0000	130.0000	47.0000	20.0000	55.0000
19	339.0505	460.0000	340.0000	60.0000	164.9495	160.0000	130.0000	59.1384	20.0000	55.0000
20	469.9999	460.0000	340.0000	117.0717	243.0000	160.0000	130.0000	76.9284	20.0000	55.0000
21	469.9827	377.4891	340.0000	60.0000	243.0000	160.0000	130.0000	47.0000	41.5282	55.0000
22	200.8316	460.0000	340.0000	60.0000	155.1684	160.0000	130.0000	47.0000	20.0000	55.0000
23	150.9585	159.5094	340.0000	60.0000	209.1636	160.0000	130.0000	47.0000	20.3686	55.0000
24	150.1325	153.0387	258.2637	60.0000	150.5651	160.0000	130.0000	47.0000	20.0000	55.0000
Total generation cost ($) = 1001746.4962										

Table 6 Comparison of results for 10-unit systems

Method	Production cost ($)	Computing time (s)
Hybrid EP-SQP [11]	1031746	NA
DGPSO [12]	1028835	NA
Hybrid PSO-SQP [13]	1027334	NA
ICA [14]	1022205.6846	NA
Proposed	1001746.4962	4.2798

NA: Not Available

VI. CONCLUSION

In this paper, a modified differential evolution has been successfully applied for solving non-convex dynamic economic dispatch problem. The method is an improved version of original DE in which the initialization it utilizes opposition-based learning, tournament best and utilizes only one set of population. The obtained results from the two test systems have indicated that the proposed MDE method has a much better performance than the other optimization methods reported in the literature. Therefore, the proposed MDE method is a promising method for online non-convex dynamic economic dispatch.

REFERENCES

[1] X. S. Han, H. B. Gooi, D. S. Kirschen, “Dynamic Economic Dispatch: feasible and optimal solutions”, IEEE Transactions on Power Systems, vol.16, no.1, pp.22-28, 2001.

[2] X. Xia, A.M. Elaiw, “Optimal Dynamic Economic Dispatch of Generation: A review”, Electric Power Systems Research, vol.80, no. 8, pp.975- 986, 2010.

[3] G. P. Granelli, P. Marannino, M. Montagna, and A. Silvestri, “Fast and Efficient Gradient Projection Algorithm for Dynamic Generation Dispatching”, IEE Proc. Gen. Trans. Distr., vol.136, pp.295-302, September 1989.

[4] K. S. Hindi and M. R. Abd Ghani, “Dynamic Economic Dispatch for Large Scale Power Systems: A Lagrangian Relaxation Approach”, Electric Power Systems Research, vol. 43, pp. 51-56, 1991.

[5] C. B. Samuah, and N. Khunaizi, “Application of Linear Programming Re-Dispatch Technique to Dynamic Generation Allocation”, IEEE Transactions on Power Systems, vol.5, pp.20-26, 1990.

[6] F. Li, R. Morgan, D Williams, “Hybrid Genetic Approaches to Ramping Rate Constrained Dynamic Economic Dispatch”, Electric Power Systems Research, vol. 43, pp. 97-103, 1997.

[7] C. K. Panigrahi, P. K. Chattopadhyay, R. N. Chakrabarti, M. Basu, “Simulated Annealing Technique for Dynamic Economic Dispatch”, Electric Power Components and Systems, vol.34, pp. 577-86, 2006.

[8] R. Balamurugan, S. Subramanian, "Differential Evolution-based Dynamic Economic Dispatch of Generating Units with Valve-Point Effects", Electric Power Components and Systems, vol. 36, pp. 828-43, 2008.

[9] R. Balamurugan, S. Subramanian, "An Improved Differential Evolution based Dynamic Economic Dispatch with Nonsmooth Fuel Cost Function", Journal of Electrical Systems, vol. 3, no. 3, pp. 151-61, 2007.

[10] R. Chakrabarti, P. K. Chattopadhyay, M. Basu, and C. K. Panigrahi,"Particle Swarm Optimization Technique for Dynamic Economic Dispatch", Institute of Engineers, vol. 87, pp. 48-54, July 2005.

[11] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa,"A Hybrid EP and SQP for Dynamic Economic Dispatch with Nonsmooth Fuel Cost Function", IEEE Transactions on Power Systems, vol. 17, no. 2, pp. 411-416, May 2002.

[12] T. Aruldoss Albert Victoire, and A. Ebenezer Jayakumar, "Deterministically Guided PSO for Dynamic Dispatch Considering Valve-Point-Effect," Electric Power Systems Research, vol. 73, no. 3, pp. 313-322, 2005.

[13] T. Aruldoss Albert Victoire, and A. Ebenezer Jayakumar, "Reserve Constrained Dynamic Dispatch of Unit with Valve-Point-Effect," IEEE Transactions on Power Systems, vol. 20, no. 3, pp. 1273-1282, August 2005.

[14] Behnam Mohammadi-ivatloo, Abbas Rabiee, Alireza Soroudi, and Mehdi Ehsan, "Imperialist Competitive Algorithm for Solving Non-convex Dynamic Economic Power Dispatch," Energy, vol. 44, pp. 228-240, 2012.

[15] R. Storn, K. Price, "Differential Evolution a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", J. Global Optim., vol. 11, no. 4, pp. 341-359, 1997.

[16] K. Price, R. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer, Berlin, Heidelberg, 2005.

[17] M. Ali, M. Pant and A. Abraham, A Modified Differential Evolution Algorithm and Its Application to Engineering Problems, 2009 International Conference of Soft Computing and Pattern Recognition, pp. 196-201, 2009.

