Industrial Engineering Journal
ISSN: 0970-2555
Volume: 54, Issue 1, January:2025

A MULTI-POPULATION-BASED DIFFERENTIAL EVOLUTION ALGORITHM DEVELOPED TO OPTIMIZE ECONOMIC LOAD DISPATCH ISSUES WITH VALVE-POINT EFFECTS

Rakesh Roshan Swain, Assistant Professor, Department of Electrical Engineering
rakeshroshnswain@gietbbsr.edu.i, GIET,Ghangapatana, BBSR
Ajanta Priyadarshine, Assistant Professor, Department of Electrical Engineering
ajantapriyadarshinee@gietbbsr.edu.in GIET,Ghangapatana, BBSR

ABSTRACT:
The valve-point effects of generation units in the economic dispatch (ED) problem make it a non-smooth and non-convex problem. This work offers an algorithm for differential evolution based on multi-population (MPDE) to handle by consideration of valve-point effects in economic load dispatch problems. Negative aspects of the conventional differential evolution algorithm are overcome by the suggested MPDE algorithm, which uses the evolutionary methods of multiple populations. Each set of populations in a multiple population has its own parameters and mutation method to improve the capacity for searching. Additionally, information sharing between various populations can boost the diversity of unique individuals within a single population. Furthermore, the technique comprises the normal distribution function to dynamically modify the scaling factor and crossover rate, hence expediting the rate of convergence. Tests of the suggested approach are conducted using the IEEE 13 unit test systems. The MPDE algorithm can achieve far fewer variations than other intelligent algorithms, according to simulation data. When it comes to solving economic dispatch problems involving valve-point effects, the proposed algorithm performs noticeably better in terms of accuracy.

Keywords—Differential Evolutions, Multi-Populations, Valve Point Effects, Economic Load Dispatch(ELD) Problems.

INTRODUCTION :

One of the most significant problems in modern computer-aided power system design is Economic Load Dispatch (ELD). The ELD problem focuses on how to distribute load among the committed producing units while meeting capacity and power balance requirements and reducing overall operating costs.[1]. The ELD problem aims to decrease the cost of power generation while optimizing the output power of each unit. Numerous measures and studies have been implemented to achieve significant cost reductions in operations.

In accordance with valve-point effects [8], the generation unit's characteristic curve deviates from linearity. Furthermore, the power system contains a large number of power producing units, which might make calculations more challenging and more likely to result in local optimal solutions. Certain mathematical techniques, such as the linear programming algorithm (LP) [9], quadratic programming algorithm (QP) [10], and dynamic programming algorithm (DP) [11], cannot effectively solve ELD problems due to their numerous non-linearity, non-convexity, and multi-dimensionality. Many researchers concentrate on heuristic intelligence optimization algorithms, such as simulated ecosystem algorithms, evolutionary algorithms that mimic the evolution of biological organisms, and swarm intelligence algorithms that mimic the behavior of biological swarms, in an effort to get around the drawbacks of traditional mathematical techniques.

The Chebyshev polynomial fitting problem led Rainer Storn and Kenneth Price to create the differential evolution algorithm (DE), an intelligent optimization method that mimics the evolution of natural organisms. Due to its powerful search capability, the DE algorithm has been enhanced by numerous researchers[12].The multi-population differential evolution algorithm (MPDE) proposed in this research utilizes distinct mutation methods for each population. The following is a list of this paper's key contributions in comparison to previous research:

1) A plan for many populations is suggested. Various combinations of factors and mutation procedures within each population will result in distinct search features. The multi-population approach deftly blends several mutation tactics to improve search performance.

2) A population-level learning technique was created.
In order to maximize individual diversity within a single population and prevent it from settling on a local optimal solution, this technique encourages information transmission between populations.

3) The scaling factor and crossover rate are dynamically adjusted using the normal distribution function to quicken the rate of convergence.
4) The MPDE algorithm can converge to the ideal value and has a lower standard deviation in the test of 13 unit test systems.

The structure of this paper is as follows: The conventional differential evolution algorithm is explained in Section II. The MPDE algorithm is proposed in Section III. The MPDE algorithm is used in Section IV to solve the ED problem. The analysis and results of the simulation are covered in Section V. Section VI provides a summary of this work's conclusion.

II. CONVENTIONAL DE ALGORITHM

Initialization, mutation operation, crossover operation, and selection operation are the four primary operational operations of the DE algorithm.

A. Initialization

The population consists of several individuals, each of whom can be viewed as a potential solution in the search space. In the event that the population contains I_N members, the population can be represented as follows:

$$N^r = \{y_m^r \mid y_m^r = (y_{m,1}^r, y_{m,2}^r, y_{m,3}^r, y_{m,4}^r, \ldots\ldots, y_{m,D}^r)^R\}, m = 1,2,3,\ldots,I_N \quad (1)$$

Where D is the number of dimensions of the individual vector, y_m^r is the m-th individual vector, N^R is the population at the r-th generation, and r is the current number of evolutions.

The technique generates the initial answers using a uniformly distributed random function in order to cover the whole search space as much as feasible for the beginning population. $y_{m,n}^0$ is computed in this way:

$$y_{m,n}^0 = y_n^{\min} + rand(0,1) \times (y_n^{\max} - y_n^{\min}) \quad (2)$$

Where (0,1) rand is a uniformly distributed random number in the interval (0,1), $y_n^{\max}$ is the maximum boundary value of the n-th dimension of the individual, $y_n^{\min}$ is the minimum boundary value of the n-th dimension of the individual and finally $y^0_{m,n}$ is the value in the n-th dimension of the individual m.

B. Operation for Mutation

Through mutation operation, the differential evolution algorithm preserves the population's diversity. The following lists the most popular DE mutation techniques [13,14]:

DE/Rand/1:

$$v_m{}^t = x^t_{r_1} + F^t_m \times (x^t_{r_2} - x^t_{r_3}) \quad (3)$$

DE/Rand/2:

$$v_m{}^t = x^t_{r_1} + F^t_m \times (x^t_{r_2} - x^t_{r_3}) + F^t_m \times (x^t_{r_4} - x^t_{r_5}) \quad (4)$$

DE/Best/1:

$$v_m{}^t = x^t_{best} + F^t_m \times (x^t_{r_1} - x^t_{r_2}) \quad (5)$$

DE/Best/2:

$$v^t_m = x^t_{best} + F^t_m \times (x^t_{r_1} - x^t_{r_2}) + F^t_m \times (x^t_{r_3} - x^t_{r_4}) \quad (6)$$

DE/current to best/1:

$$v^t_m = x^t_m + F^t_m \times (x^t_{best} - x^t_m) + F^t_m \times (x^t_{r_1} - x^t_{r_2}) \quad (7)$$

DE/rand to best/1:

$$v^t_m = x^t_{r_1} + F^t_m \times (x^t_{best} - x^t_{r_1}) + F^t_m \times (x^t_{r_2} - x^t_{r_3}) \quad (8)$$

C. Crossover Operation

The DE algorithm carries out the crossover operation based on x^t_m, whereas t generates t through mutation operation and t to produce the trial vector t. Examine the random number and the crossover rate to produce each trial vector dimension. The equation for updating is provided by:

$$u^t_{m,n} = \begin{cases} v^t_{m,n} & \text{If } rand(0,1) \le CR^t_m \text{ or } n = n_{rand} \\ x^t_{m,n} & \text{Otherwise} \end{cases} \quad (9)$$

where (0,1) *rand* is a uniformly distributed random number in the interval (0,1) ; n_{rand} is a random integer in the interval [1,] D ; $CR_m{}^t$ is the crossover rate of individual m at the t-th generation.

D. Selection Operation

The selection process is carried out by the DE algorithm using the greedy selection strategy. The vector with the higher fitness value is chosen as the next generation of individuals after comparing the values of the fitness function corresponding to the x^t_m and v^t_m. Consequently, the following is a definition of the selection operation:

$$x_m^{t+1} = \begin{cases} u_m^t & f(u_m^t) \le f(x_m^t) \\ x_m^t & \text{Otherwise} \end{cases} \quad (10)$$

III. MPDE ALGORITHM:

This work proposes a multi-population (MPDE)-based differential evolution algorithm that outperforms the conventional DE algorithm in two ways.

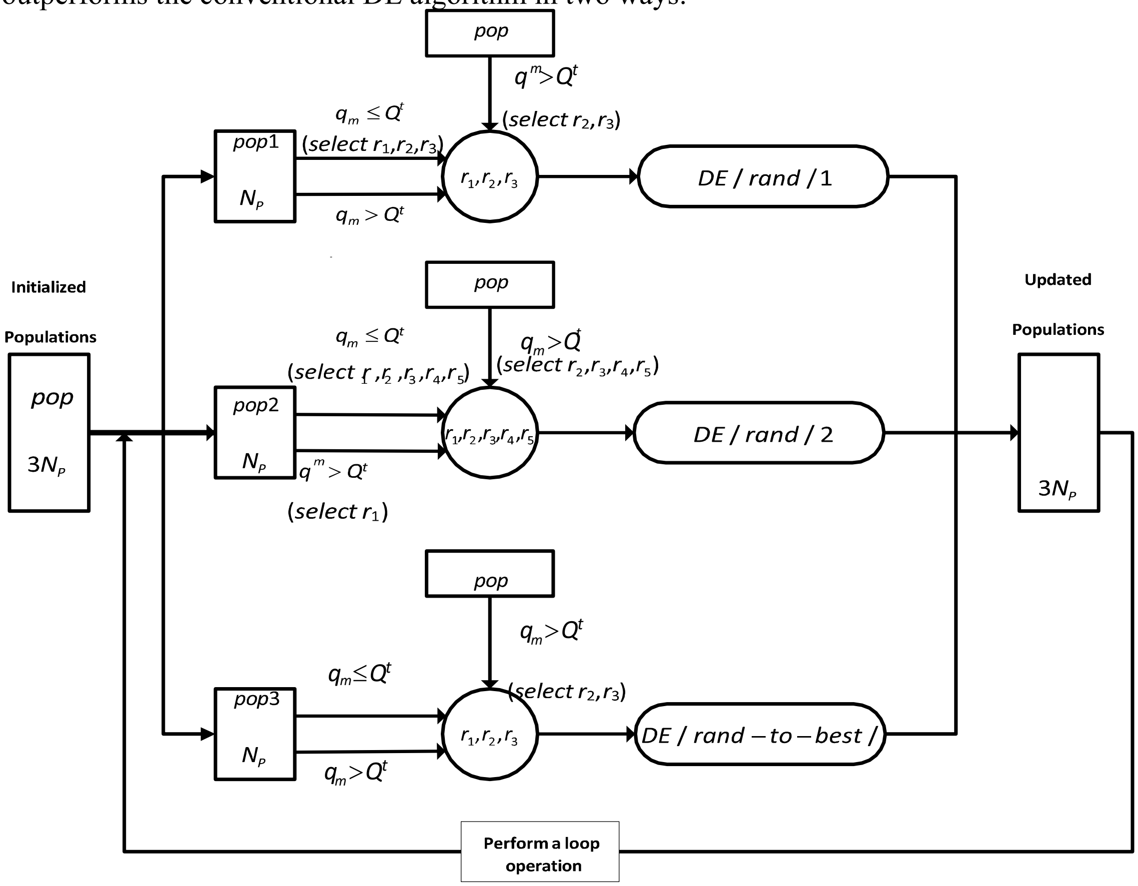

FIGURE 1. The schematic diagram of the MPDE algorithm

A. MULTI POPULATION CO-EVOLUTION
   Multi-population co-evolution is the fundamental component of the MPDE method. For the DE algorithm, a multi-population approach is developed in this research, with distinct mutation techniques used by each population. Multiple populations can share knowledge and learn from one another, increasing the diversity of the single population.
B. DYNAMICALLY ADJUST ALGORITHM PARAMETER

The DE algorithm's global search capability and convergence speed are significantly impacted by the scaling factor and crossover rate. The command of the diversity of parameters in the standard DE algorithm is somewhat low, and the effect of fixed parameters is not very effective for certain special issues because the parameters are chosen from pre-set values. The control parameters are changed from fixed to dynamic in this paper's proposed method of dynamic parameter adjustment.

IV. APPLICATION OF MPDE ALGORITHM

The MPDE algorithm is proposed in section III. This section presents the use of the MPDE algorithm for ED issue solving. Among its primary contents is the goal function, limitations, and handling of the ED problem. Furthermore, the precise procedures for using the MPDE algorithm to solve the ED problem are presented.

The ED's goal is to minimize the overall cost of power generation by optimizing the generator unit's power output while meeting the power system's limits. When taking valve-point effects into account, the cost function can be expressed as follows:

A. CONSTRAINTS:

1. Unit Power In equality constraints :

$$P_j^{\max} \geq P_j \geq P_{\leq j}^{\min} \quad (11)$$

2. Ramp Rate Constraints :

$$p_j - p_j^0 \leq UR_j \text{ and } p_j^0 - p_j \leq DR_j \quad (12)$$

3. Prohibited Operating zone constraints:

$$P_j \in \begin{cases} P_j^{\min} \leq P_j \leq P_{j,1}^l \\ P_{j,k-1}^u \leq P_j \leq P_{j,k}^l, k = 2,3,\dots,g_j \\ P_{j,g_j}^u \leq P_j \leq P_j^{\max} \end{cases} \quad (13)$$

4. System power Equality Constraints:

$$\sum_{j=1}^{M} P_j = P_D + P_{Loss}$$

$$P_{Loss} = \sum_{j=1}^{M}\sum_{i=1}^{M} P_j B_{ji} P_i + \sum_{j=1}^{M} B_{0j} P_j + B_{00} \quad (14)$$

B. CONSTRAINTS HANDLING:

1. INEQUALITY RAMP RATE CONSTRAINT HANDLING

Following the crossover operation, the algorithm may produce a new individual vector that does not meet the criteria of inequality and ramp-rate. When this occurs, the altered each generating unit's output power is computed as follows:

$$P_j = \begin{cases} \max(P_j^{\min}, P_j^0 - DR_j) & P_j \leq \max(P_j^{\min}, P_j^0 - DR_j) \\ \min(P_j^{\max}, P_j^0 + UR_j) & P_j \geq \min(P_j^{\max}, P_j^0 + UR_j) \\ P_j & \text{Otherwise} \end{cases} \quad (15)$$

2. PROHIBITED OPERATING ZONE CONTRAINT HANDLING

The output power of each generator unit is modified as follows if the generator units of the new individual vector produced by the algorithm are located in areas where operation is prohibited:

$$P_j = \begin{cases} P_{j,k}^{l} & \text{If}(P_j - P_{j,k}^{l}) \le (P_{j,k}^{u} - P_j) \\ P_{j,k}^{u} & \text{Otherwise} \end{cases} \quad (16)$$

3. EQUALITY CONSTRAINT HANDLING

In this manner, the generator unit can be adjusted to meet equality criteria while also reducing the influence caused by output power variance and altering the generator unit's output power less. The following is an analysis of the particular steps:

Step-1 :
If violates formula(24)or satisfies formula(25),set the transition variable, else $T_j = 0$ $T_P = T_j$.
Step-2 :
Calculate the difference Δ between the current total output and the demand output.

$$\Delta = \sum_{j=1}^{M} P_j - P_D - P_{Loss} \quad (17)$$

Step-3 :
Modify the output of P_j in order to satisfy the equality constraints (26) with following formula :

$$P_j = P_j - \Delta \times \frac{T_j}{\sum_{j=1}^{M} T_j} \quad (18)$$

Step-4 :
Check all the modified P_j, if there is any violation of the inequality constraints, Perform Formula (15)and (16),back to step1.

C. STEPS OF APPLYING MPDE ALGORITHMS TO ED PROBLEMS:

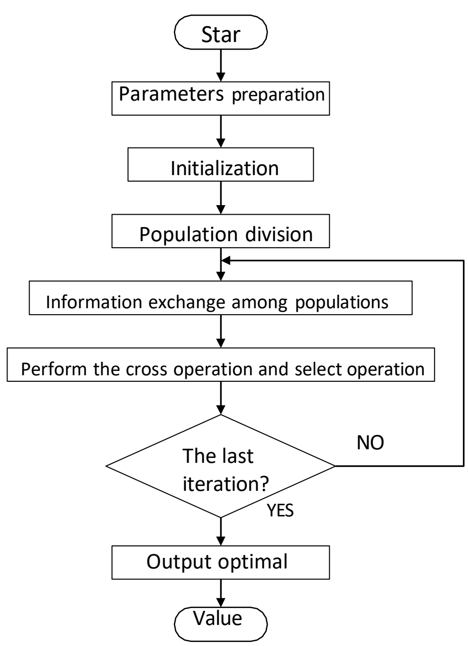

FIGURE2.The flow chart of MPDE algorithm in solving ED problem

V. SIMULATION AND RESULT ANALYSIS:

To verify the all-round performance of the improved algorithm, we test here cases of 13 unit test systems using this algorithm. All the cases are coded in C++ and implemented in Visual Studio 2013, which is tested on a PC with an Intel i5 2.3GHz processor, 4GB of RAM and Windows 10 Professional, each case runs 50 times independently, and we compare them with the results of other intelligent algorithms.

A. Setting Algorithm Parameter:

population	□min	□max	□min	□max
*pop*1	0.7	1	0.1	0.6
*pop*2	0	0.3	0	0.6
*pop*3	0.6	0.9	0	0.6

Table -1 : The parameters of the MPDE algorithm for solving the unit ED problems

Cases	Case1	Case2	Case3	Case4	Case5	Case6
M	13	13	40	40	80	140
VPE	√	√	√	√	√	√
TL	X	√	X	√	X	X
POZs	X	X	X	X	√	X
RRL	X	X	X	X	X	√
P_D (MW)	1800	2520	10500	10500	21000	49342
NP	90	90	240	240	240	240
t_{max}	12000	10000	10000	10000	20000	20000

Table – 2: Input parameters and the brief introduction to test cases.

B. For 13 – Unit Test System :

One example of the cases in 13- unit test system, which includes the valve-point effects and its load demand is 1800 MW. Two sets of data are used for fuel consumption cost coefficients. In this case, One set of the data of fuel consumption cost coefficients and generation limits were referred from [44]. The Second set, DataSet2 (13-unit), for fuel consumption cost coefficients and generations limits refer to [4]. The difference between them is the E fuel consumption cost coefficient of 3-th unit. Case 1 was run independently for 50 times with the MPDE algorithm. Figure 4 shows the convergence characteristics of the MPDE algorithm when solving case 1. Table III shows the output of each generator unit at the

Lowest total generation cost with the different fuel consumption cost coefficients.

Unit	DataSet1(13-u	DataSet2(13-unit)
	Output (MW)	Output (MW)
1#	628.3185307	628.3185307
2#	149.5996502	222.7490688

Industrial Engineering Journal
ISSN: 0970-2555
Volume: 54, Issue 1, January:2025

3#	222.7490688	149.5996502
4#	109.8665501	109.8665501
5#	109.8665501	109.8665501
6#	109.8665501	109.8665501
7#	109.8665501	109.8665501
8#	109.8665501	109.8665501
9#	60	60
10#	40	40
11#	40	40
12#	55	55
13#	55	55
Load (MW)	1800	1800
Cost($/h)	17960.366122	17963.829201

Table 3 : Output power of the generator for the best result for case 1

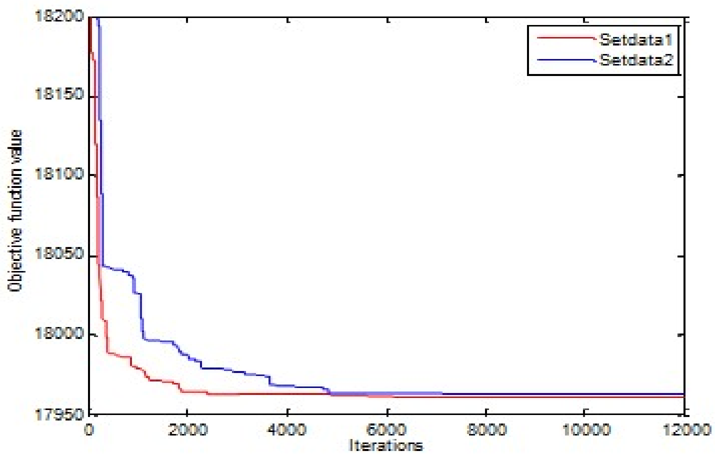

Figure 3 : The convergence characteristics for 13 Unit Bus System

DataSet1(13-unit)						
Methods	Minimum($/h)	Maximum($/h)	Mean($/h)	Std. dev	Time(s)	NFE
DE [45]	17968.3601	18133.4582	18002.9099	38.3352	10.5	25*1000
SOMA [46]	17967.4219	18017.6161	17985.3242	20.6772	–	25*1000
CDE [45]	17967.4	18065.8044	17995.585	27.0900	12.1	25*1000
HS [47]	17965.6204	18070.1762	17986.5626	26.3702	–	22500
HQPSO [44]	17963.9571	18633.6435	18273.861	123.2242	–	20*800
DEC-SQP [48]	17963.94	17984.81	17973.13	–	0.34	30*600

ABC [49]	17962.4279	–	–	–	–	–
CDEMD [45]	17961.944	18061.411	17974.6869	20.3066	12.6	25*1000
MDE [32]	17960.39	17969.09	17967.19	–	–	80*3500
HCR-DE [34]	17960.38	17961.04	17960.59	0.069	4.91	26*100
IPSO-TVAC [50]	17960.3703	17961.273	17960.641	–	–	100*200
ICA-PSO [51]	17960.37	17978.14	1797.94	–	–	–
SDE [37]	17960.37	–	–	–	–	60*300
THS (t=2s) [52]	17960.37	–	17982.98	–	–	100*5000
THS (t=5s) [52]	17960.37	–	17985.15	–	–	100*5000
THS (t=8s) [52]	17960.37	–	17977.60.	–	–	100*5000
IDE [36]	17960.3661	17969.4857	17961.4717	2.6499	7.535	150*3000
HIS [47]	17960.3661	17971.6512	17965.4152	16.9531	–	22500
CSOMA [46]	17960.3661	17970.8323	17967.8708	0.8858	–	25*1000
DHS [53]	17960.3661	17968.3610	17961.1226	1.92	0.12	50*1200
MPDE	17960.3661	17960.5044	17960.3716	0.027	3.0	90*12000
DataSet2(13-unit)						
CEP [54]	18048.21	18190.32	18404.04	–	–	–
IFEP [54]	17994.07	18267.42	18127.06	–	–	–
EP-SQP [55]	17991.03	–	18106.93	–	121.93	100*100
NDS [56]	17976.95	17976.95	17976.95	–	1.5634	–
CGA_MU [57]	17975.34	–	–	–	27.91	–
TLBO [58]	17972.81	18243.12	18080.87	–	–	60*800
PSO-SQP [55]	17969.93	–	18029.99	–	33.97	100*100
CASO [59]	17965.15	–	18022.04	–	22.19	–
FCASO-SQP [59]	17964.08	–	18001.96		19.62	–
IGA_MU [57]	17963.9848	–	–	–	8.27	–
ST-HDE [37]	17963.89	–	18046.38	–	1.41	NA*2500
CE-SQP [60]	17963.85	–	17965.97	–	34.33	–
FAPSO-NM [61]	17963.84	17964.21	17963.9577	–	6.8	26*300
NSEO [62]	17963.8346	18186.9043	18052.7191	32.29	0.16	–
CBA [18]	17963.83	17995.2256	17965.4889	6.8473	0.97	40*300
DE [5]	17963.83	17975.36	17965.48	–	–	78*1200
UHGA [63]	17963.83	–	17988.04	–	8.48	28*30
BA [61]	17963.83	18288	18085.06	–	–	78*1200
MsEBBO [23]	17963.8292	17969.0323	17964.0468	1.9215	–	80*1000
MABC [64]	17963.8292	17963.8305	17963.8293	0.0002	38.2	12*18000
FAPSO–VDE [35]	17963.8292	17963.832	17963.828	–	4.1	26*100

MPDE	17963.8292	17963.8292	4 17963.829 2	0	3.0	90*12000

V. CONCLUSION

Here we present an ED problem for the same with valve point effects, but using differential evolution as its basis and also showing a multi-population based on this algorithm. In order to create MPDE, different mutations or parameters must be present in each MPDEA as it evolves. An essential component is present in MPDE. Because information flow can be carried between and around any single population of differential evolved populations, the algorithm's ability to learn solutions from an individual difference is much faster than traditional Differential Evolution Algorithm. Moreover, the normal distribution function in MPDE algorithm is used to dynamically change the scaling factor and crossover rate. MPDE algorithm is tested using the 13-, 40–4, 80–, and 140– unit test systems.. The MPDE algorithm is deemed more accurate and robust than other intelligent algorithms, with statistical evidence that it can offer adequate global optimization solutions. The MPDE algorithm is a suitable solution for solving the valve point effects of the ED problem. This conclusion suggests that it is an effective tool.

REFERENCES :

[1] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading," *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1325-1332, Aug. 1993.

[2] R. A. Jabr, A. H. Coonick and B. J. Cory, "A homogeneous linear programming algorithm for the security constrained economic dispatch problem," *IEEE Transactions on Power Systems*, vol. 15, no. 3, pp. 930-936, Aug. 2000.

[3] Ji-Yuan Fan and Lan Zhang, "Real-time economic dispatch with line flow and emission constraints using quadratic programming," *IEEE Transactions on Power Systems*, vol. 13, no. 2, pp. 320-325, May 1998.

[4] 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, Feb. 2003.

[5] Zwe-Lee Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1187-1195, Aug. 2003.

[6] Song, Y.H.; Chou, C.S.; Stonham, T.J. "Combined heat and power economic dispatch by improved ant colony search algorithm," *Electric Power Systems Research*, vol 52, no 2, pp 115-121, November 1, 1999.

[8] Po-Hung Chen and Hong-Chan Chang, "Large-scale economic dispatch by genetic algorithm," *IEEE Transactions on Power Systems*, vol. 10, no. 4, pp. 1919-1926, Nov. 1995.

[9] K. P. Wong and C. C. Fung, "Simulated annealing based economic dispatch algorithm," *IEE Proceedings C - Generation, Transmission and Distribution*, vol. 140, no. 6, pp. 509-515, Nov. 1993.

[10] R. Naresh, J. Dubey, J. Sharma, "Two-phase neural network based frame work for modelling of constrained economic load dispatch," I*EE Proc. Generation, Transmission and*

Distribution, vol. 151, no. 3,
pp. 373–378, May 2004.

[11] W. M. Lin, F. S. Cheng and M. T. Tsay, "An Improved Tabu Search for Economic Dispatch with Multiple Minima," *IEEE Power Engineering Review*, vol. 22, no. 1, pp. 70-70, Jan. 2002.

[12] B. K. Panigrahi and V. R. Pandi, "Bacterial foraging optimisation: Nelder-Mead hybrid algorithm for economic load dispatch," *IET Generation, Transmission & Distribution*, vol. 2, no. 4, pp. 556-565, July 2008.

[13] G. P. Singh and A. Singh, "Comparative study of krill herd, firefly and cuckoo search algorithms for unimodal and multimodal optimization," *Int. J. Intell. Syst*. Appl., vol. 6, no. 3, pp. 35−49, Feb. 2014.

[14] R. Kumar, A. Sadu, R. Kumar, S. K. Panda, "A novel multi-objective directed bee colony optimization algorithm for multi-objective emission constrained economic power dispatch", *Int. J. Electric. Power Energy Syst*, vol. 43, no. 1, pp. 1241−1250, December. 2015.

[15] Roy, P. K, S. P. Ghoshal, and S. S. Thakur. "Biogeography-based Optimization for Economic Load Dispatch Problems," *Electric Power Components and Systems*, vol. 38, no. 2, pp. 166−181, December. 2009.

[16] J. Zhao, S. Liu, M. Zhou, X. Guo and L. Qi, "Modified cuckoo search algorithm to solve economic power dispatch optimization problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 4, pp. 794-806, July 2018.

[17] Cai Jiejin, Ma Xiaoqian, Li Lixiang, and Peng Haipeng, "Chaotic particle swarm optimization for economic dispatch considering the generator constraints," *Energy Conversion and Management*, vol. 48, no. 2, pp 645-653, February 2007.

[18] Cai Jiejin, Ma Xiaoqian, Li Lixiang, and Peng Haipeng, "Chaotic particle swarm optimization for economic dispatch considering the generator constraints," *Energy Conversion and Management*, vol. 48, no. 2, pp 645-653, February 2007.

[19] B. R. Adarsh, T. Raghunathan, T. Jayabarathi, and X. S. Yang, "Economic dispatch using chaotic bat algorithm," *Energy*, vol. 96, no. 1, pp. 666− 675, Feb. 2016.

[20] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Transactions on Power Systems*, vol. 20, no. 1,
pp. 34-42, Feb. 2005.

[21] V.K. Jadoun, N. Gupta, K.R. Niazi, A. Swarnkar, "Modulated particle swarm optimization for economic emission dispatch", *International Journal of Electrical Power and Energy Systems*, vol 73, pp. 80-88, December 1, 2015.

[22] K. T. Chaturvedi, M. Pandit and L. Srivastava, "Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch," *Int. J. Electr. Power Energy Syst*., vol. 31, no. 6, pp. 249–257, July. 2009.

[23] He, Dakuo; Wang, Fuli; Mao, Zhizhong, "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect," *International Journal of Electrical Power and Energy Systems*, vol 30, no 1, pp 31-38, January 2008.

[24] Xiong, Guojiang; Shi, Dongyuan; Duan, Xianzhong, "Multi-strategy ensemble biogeography-based optimization for economic dispatch problems", *Applied Energy,* vol 111, pp 801-811, November 2013.

[25] Rainer Storn,Kenneth Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol 11, no 4, pp 341-359, December 1997

[26] A. Draa, S. Bouzoubia, I. Boukhalfa, "A sinusoidal differential evolution algorithm for

numerical optimization," *Applied Soft Computing*, vol 27, pp 99-126, February 2015.

[27] A. W. Mohamed, "An improved differential evolution algorithm with triangular mutation for global numerical optimization," *Computers & Industrial Engineering*, vol 85, pp 359-375, July 2015.

[28] G.H. Li, Q.Z Lin, L.Z. Cui, Z.H. Du, Z.P. Liang, J.Y Chen, N. Lu, Z. Ming, "A novel hybrid differential evolution algorithm with modified CoDE and JADE," *Appl Soft Computing*. vol 47, pp 577–599, October 2016.

[29] Mohamed, A.W. & Mohamed, A.K. "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol 10, pp 253–277, February 2019.

[30] Laizhong Cui, Genghui Li, Zexuan Zhu, Qiuzhen Lin, Ka-Chun Wong, Jianyong Chen, Nan Lu, Jian Lu, "Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism," *Information Sciences*, vol 422, pp 122-143, January 2018,

[31] Guohua Wu, Xin Shen, Haifeng Li, Huangke Chen, Anping Lin, P.N. Suganthan, "Ensemble of differential evolution variants," *Information Sciences*, vol 423, pp 172-186, January 2018.s