Economic Dispatch in Microgrid using Stochastic Fractal Search Algorithm

Thang Phan Van Hong and Tung Tran The

Abstract— *This paper presents the method using Stochastic Fractal Search Algorithm (SFSA) for solving the Economic dispatch (ED) problem in Microgrid. The SFSA is based on two main processes: diffusion process and updating process. The diffusion process is similar to Fractal Search about diffusion of new particle, but only the best generated particle from the diffusing process is considered, and the others are discarded. It uses Levy flight and Gaussian statistical methods to control new particles creation process. In the updating process, using the algorithm which simulates how a point in the group updates its position based on the position of other points in the group, through random methods. The proposed method has been applied on 3 problems in microgrid system: System with 140 diesel generators and solved fuel constrained economic dispatch (ED) problem; system with 40 generators with valve-point loading; system with two conventional generators (synchronous generators), one combined heat and power (CHP), wind generator, solar generator - in the islanded mode. The objective of the ED problem is to minimize the total generation cost of a power system over some appropriate period while satisfying various constraints. This paper gives effective solutions for these 3 problems and compares their results with those obtained by other evolutionary methods. It is found that Stochastic Fractal Search Algorithm is able to provide better solution.*

***Keywords* — Stochastic fractal search algorithm, microgrid, economic dispatch.**

1. INTRODUCTION

Microgrids are electricity distribution systems containing loads and distributed energy resources,(such as distributed generators, storage devices, or controllable loads) that can be operated in a controlled, coordinated way either while connected to the main power network or while is landed. With the diversity of the input energy, microgrid system has an abundance of cost-consuming problems. Two prioritized issues need research is power dispatch and economic dispatch in the system. Economic dispatch is distribution modes of inputs to ensure power supply to the system load in an optimal way with the lowest cost. The cost optimal must ensure load capacity is stable and reliable.

Economic dispatch problem of thermal generating units with non-smooth/non-convex cost functions due to valve-point loading taking into account transmission losses and nonlinear generator constraints such as prohibited operating zones [1].

The economic dispatch problem in microgrid system was solved with different methods from traditional to artificial intelligence-based. Many researchers used conventional methods for solving EA problem in microgrid system as: Seon-Ju Ahn, Soon-Ryul Nam; Joon-Ho Choi; Seung-Il Moon [2], Florian Dorfler, John Simpson-Porco; Francesco Bullo [3], Anderson Hoke[4]….The conventional methods are all greedy search algorithm. They are easy to be implemented and provide high searching efficiency, but generally cannot converge to the global optimum solution in the large-scale distribution systems.

For the optimization problems in microgrid, with inherent complexity properties of this system, the traditional method is difficult to meet the demands of researchers. Therefore artificial intelligence search methods have become popular for solving the ED problem. They can direct searching processes to the global optimum at the probability of one hundred percent theoretically. However, they all inevitably involve a large number of computation requirements and control parameters.

The ED problem in microgrid is solved effectively by adopting artificial intelligence methods such as Chaotic quantum genetic algorithm (CQGA) [5], PSO algorithm combined with Monte Carlo simulations [6], Cuckoo search algorithm (CSA) [7], Isolation Niche and Immune Genetic Algorithm (INIGA) [8], Particle Swarm Optimizers (PSO) [9], Improved differential evolution (IDE) [10], Dynamic programming algorithm (DPO) [11], Model Predictive Control (MPC) [12].

The property of an object or quantity which explains self-similarity on all scales, in a somewhat technical sense, is called fractal. The term of ‘‘fractal’’ comes from the Latin word *fractus* which means ‘‘broken’’ or ‘‘fractured’’, and it was first used by Benoıt Mandelbrot in 1975. Mandelbrot also tried to use the concept of fractal theories to describe geometric patterns in nature [13]. Based on the fractal characteristics, new meta-heuristic method inspires random fractals grown by Diffusion Limited Aggregation (DLA) method concept as a successful search algorithm in both accuracy and time consumption.

According to fractal growth (DLA method) and

Thang Phan Van Hong and Tung Tran The (corresponding author) are with Department of Power Systems, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam. Tel: +84-8-38657 296 – Ext: 5729; Email: tungqna@gmail.com.

potential theory loosely speaking, Fractal Search was introduced by Hamid Salimi and his coordinators. It uses three simple rules to find a solution:

1. Each particle has an electrical potential energy.
2. Each particle diffuses, and causes some other random particles to be created, and the energy of the seed particle is divided among generated particles.
3. Only few of the best particles remain in each generation, and the rest of the particles are disregarded.

In this paper, an SFSA is proposed for the ED in microgrid problem. The SFSA is developed by Hamid Salimi and his coordinators [14] in 2013. Based on some disadvantages of FSA, another version of Fractal Search Algorithm called Stochastic Fractal Search Algorithm (SFSA) was introduced.

The purpose of this paper is to apply the SFSA to solve the ED in microgrid problem. The proposed method has been tested on 3 cases, including: 140 diesel generators [10] – 40 generators with valve point loading [15] – 2 generators, 1 combined heat and power (CHP),1 wind generator and 1 solar generator [16] with loop count N = 50, 500.

The obtained results are compared to those algorithm from Continuous quick group search optimizer (CQGSO), Improved-particle swarm optimization (IPSO), Differential evolution based on truncated Lévy-type flights and population diversity measure (DEL) [10] - Classical evolutionary programming (CEP), Fast evolutionary programming (FEP), evolutionary programming MFEP, and Improved FEP (IFEP) [15] - Reduced Gradient Method (RGM) [16].

2. PROBLEM FORMULATION

The objective of the ED is to minimize the total generation cost of a power system over some appropriate period while satisfying various constraints. The practical non-smooth/non-convex ED problem considers generator nonlinearities such as valve-point loading effects, prohibited operating zones and multi-fuel options along with system power demand, transmission loss and operational limit constraints.

2.1. Economic dispatch problem considering prohibited operating zones and transmission losses

The ED problem can be described as a minimization process with the objective [10]:

$$\text{Min } \sum_{i=1}^{N_t} F_i(P_i) = \sum_{i=1}^{N_t} a_i + b_i P_i + c_i P_i^2 \quad (1)$$

where $F_i(P_i)$ is the fuel cost function of i^{th} unit; a_i, b_i and c_i are the fuel cost coefficients of i^{th} unit; N_t is the number of committed units; P_i is the power output of i^{th} unit.

The constraints of objective function are:

(i) Power balance constraint:

$$\sum_{i=1}^{N_t} P_i + P_D - P_L = 0 \quad (2)$$

The transmission loss P_L may be expressed by using B-coefficients as:

$$P_L = \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} P_i B_{ij} P_j + \sum_{i=1}^{N_t} B_{0i} P_i + B_{00} \quad (3)$$

where P_D is the system load demand. B_{ij}, B_{0i} and B_{00} are B-coefficients.

(ii) Generation capacity constraints

The power generated by each unit should be within its lower limit P_i^{min} and upper limit P_i^{max} so that:

$$P_i^{min} \le P_i \le P_i^{max} \;\; i \in N_t \quad (4)$$

(iii) Prohibited operating zone

The feasible operating zones of a unit with prohibited operating zones can be described as follows:

$$\begin{aligned} &P_i^{min} \le P_i \le P_{i,1}^{l} \\ &P_{i,j-1}^{u} \le P_i \le P_{i,j}^{l},\; j = 2,3, \dots n_i \\ &P_{i,n_i}^{u} \le P_i \le P_i^{max} \;\; i \in N_t \end{aligned} \quad (5)$$

where j represents the number of prohibited operating zones of i the unit. $P_{i,j-1}^{u}$ is the upper limit of *(j-1)*th prohibited operating zone of i the unit. $P_{i,j}^{u}$ is the lower limit of jth prohibited operating zone of i the unit. Total number of prohibited operating zone of i the unit is n_i.

(iv)Calculation of slack generator

N committed generating units deliver their power output subject to the power balance constraint (2) and the respective capacity constraints (4). Assume that the power loading of first (N_t -1) generators are known, the power level of the N_t^{th} generator (i.e., the slack generator) is given by:

$$P_N = P_D + P_L - \sum_{i=1}^{N_t-1} P_i \quad (6)$$

The transmission loss P_L is a function of all generator outputs including the slack generator and it is given by:

$$P_L = \sum_{i=1}^{N_t-1} \sum_{j=1}^{N_t-1} P_i B_{ij} P_j + 2P_{N_t}\left(\sum_{i=1}^{N_t-1} B_{N_t i} P_i\right) + B_{N_t N_t} P_{N_t}^2 + \sum_{i=1}^{N_t-1} B_{0i} P_i + B_{0N_t} P_{N_t} + B_{00}) \quad (7)$$

Expanding and rearranging, Eq. (6) becomes:

$$B_{N_t N_t} P_{N_t}^2 + \left(2 \sum_{i=1}^{N_t-1} B_{N_t i} P_i + B_{0N_t} - 1\right) P_{N_t} + \left(P_D + \sum_{i=1}^{N_t-1} \sum_{j=1}^{N_t-1} P_i B_{ij} P_j + \sum_{i=1}^{N_t-1} B_{0i} P_i - \sum_{i=1}^{N_t} P_i + B_{00}\right) = 0 \quad (8)$$

The loading of the slack generating unit (i.e., N_t^{th}) can then be found by solving Eq. (8) using standard algebraic method.

2.2. Economic dispatch problem considering valve-point effects

The fuel-cost function considering valve-point loadings of the generating units is given as [15]:

$$f(P_j) = a_j P_j^2 + b_j P_j + c_j + \left| e_j \times \sin(f_j \times (P_{imin} - P_j)) \right| \tag{9}$$

where a_j, b_j, and c_j are the fuel-cost coefficients of the j^{th} unit, and e_j and f_j are the fuel cost-coefficients of the j^{th} unit with valve-point effects.

2.3. Economic Dispatch considering renewable energy cost functions

The cost function of wind generation is as [16]:

$$F(P_w) = aI^p P_w + G^E P_w \tag{10}$$

$$a = \frac{r}{[1-(1+r)^{-N}]}$$

where,

P_w: Wind generation (kW).

a: Annuitization coefficient (dimensionless).

r: Interest rate (taken as 0.09 for base case).

N: Investment lifetime (taken as N = 20 years).

I^p: Investment wind costs, per unit installed power ($/kW).

G^E: Operation & maintenance wind costs, per unit generated energy ($/kW).

The cost function of solar generation is as [16]:

$$F(P_s) = aI^p P_s + G^E P_s \tag{11}$$

$$a = \frac{r}{[1-(1+r)^{-N}]}$$

where,

P_s: Solar generation (kW)

a: Annuitization coefficient (dimensionless)

r: Interest rate (taken as 0.09 for base case)

N: Investment lifetime (taken as N = 20 years)

I^p: Investment solar costs, per unit installed power ($/kW)

G^E: Operation & maintenance solar costs, per unit generated energy ($/kW).

3. STOCHASTIC FRACTAL SEARCH ALGORITHM

3.1. The processes of stochastic Fractal Search Algorithm [14]

Two main processes that occurr in the SFSA are: The diffusing process and the updating process. In the first process, similar to Fractal Search, each particle diffuses around its current position to satisfy intensification (exploitation) property. Unlike the diffusing phase in FS which causes a dramatic increase in the number of participating points, we consider a static diffusion process for SFSA. It means that the best generated particle from the diffusing process is the only particle that is considered, and the rest of the particles are discarded. In addition to efficient exploration of the problem space, SFSA uses some random methods as updating processes. In other word, updating process in SFSA leads us to diversification (exploration) properties in metaheuristic algorithms.

To create new particles from the diffusion procedure, two statistical methods called Levy flight and Gaussian are investigated. Preliminary studies over taking advantage of Levy and Gaussian distributions separately show, however, that although Levy flight converges faster than Gaussian walk in a few generations; Gaussian walk is more promising than Levy flight in finding global minima. Therefore, unlike Fractal Search which uses the Levy flight distribution, Gaussian distribution is the only random walk to be employed in the DLA growth process of SFSA.

3.1.1 Diffusion process

A series of Gaussian walks participating in the diffusion process have been listed in the following equations:

$$GW_1 - Gaussian\ (\mu_{BP}, \sigma) + (\varepsilon \times BP - \varepsilon' \times P_i) \tag{12}$$

$$GW_2 - Gaussian\ (\mu_P, \sigma) \tag{13}$$

where ε and ε' are uniformly distributed as random numbers restricted to [0,1]. BP and P_i are denoted as the positions of the best point and the ith point in the group, respectively. The first two Gaussian parameters are μ_{BP} and σ where μ_{BP} is exactly equal to BP. The two the latter parameters are μ_P and σ where μ_P is equal to P_i. With consideration of Gaussian parameters, the standard deviation is computed by Eq. (14):

$$\sigma - \left| \frac{log(g)}{g} \times (P_i - BP) \right| \tag{14}$$

To encourage a more localized search as individuals, and get closer to the solution, the term $\frac{\log(g)}{g}$ is used in order to decrease the size of Gaussian jumps, as the number of generation increases.

Assume a global optimization problem with dimension D is at hand. Therefore, each denoted individual considered to solve the problem has been built based on a D dimensional vector. During the initialization process, each point is initialized randomly based on problem constrains by prescribing minimum and maximum bounds. The initialization equation of the j^{th} point, P_j, is addressed as follows:

$$P_j - LB + \varepsilon \times (UB - LB) \tag{15}$$

where LB and UB are the lower and the upper problem constrained vectors, respectively. As stated in previous equations, ε is a uniformly distributed random number which is restricted to [0,1] continuous area. After initializing all points, the fitness function of each point is computed to attain the best point (*BP*) among all points. According to the exploitation property in the diffusion procedure, all points have roamed around their current position to exploit problem search space. On the other hand, two statistical procedures aimed to increase the better space exploration are considered due to the exploration property. The first statistical procedure performs on each individual vector index, and the second statistical method is then applied to all points.

3.1.2 The first statistical procedure

For the first statistical procedure, at first, all the points are ranked based on the value of the fitness function. Each point *i* in the group is then given a probability value which obeys a simple uniform distribution as following equation:

$$Pa_i = \text{rank}\ (P_i)/N \tag{16}$$

where $rank(P_i)$ is consider as the rank of point P_i among the other points in the group, and N is used as the number of all points in the group. In fact, Eq. (16) wants to state that the better the point, the higher the probability. This equation is used to increase the chance of changing the position of points which have not obtained a good solution. On the other hand, the chance of passing good solutions in the next generation will increase. For each point P_i in group based on whether or not the condition $Pa_i < \varepsilon$ is satisfied, and where ε is a uniform random number belonging to [0,1], the *j*th component of P_i, is updated according to Eq. (17), otherwise it remains unchanged.

$$P_i'(j) = P_r(j) - \varepsilon \times (P_t(j) - P_i(j)) \tag{17}$$

where P_i' is the new modified position of P_i, P_r and P_t are the randomly selected points in the group, ε is the random number selected from the uniform distribution in the continuous space [0,1].

3.1.3 The second statistical procedure

Regarding to the first statistical procedure which is carried out on the components of the points, the second statistical change is aimed to change the position of a point considering the position of other points in the group. This property improves the quality of exploration, and it satisfies the diversification property. Before starting the second procedure, once again, all points obtained from the first statistical procedure are ranked based on Eq. (16). Similar to the first statistical process, if the condition $\text{Pa}_\text{i} < \varepsilon$ is held for a new point P_i', the current position of P_i' is modified according to Eqs. (18) and (19), otherwise no update occurs.

Eqs. (15) and (16) are presented as follows:

$$P_i'' = P_i' - \hat{\varepsilon} \times (P_t' - BP) \mid \varepsilon' \le 0.5 \tag{18}$$

$$P_i'' = P_i' + \hat{\varepsilon} \times (P_t' - P_r') \mid \varepsilon' > 0.5 \tag{19}$$

where $P_r'\ and\ P_t'$ are the two randomly selected points obtained from the first procedure, and $\hat{\varepsilon}$ are random numbers generated by the Gaussian Normal distribution. The new point P_i'' is replaced by P_i' if its fitness function value is better than P_i'.

3.2. Stochastic Fractal Search Algorithm [14]

3.2.1 The overall procedure of the proposed SFSA for solving the ED problem is addressed as follows:

Step 1: Initialize a population size (*N* points).

Step 2: Calculate fitness function to find the best point (BP).

Step 3: Compare value of G with maximum iteration. If G is less than maximum iteration goes to step 7. Otherwise go to Step 4.

Step 4: Call Diffusion Process.

Step 5: Call Updating Process.

Step 6: Return Step 3.

Step 7: Post process results and visualization.

3.2.2 The overall procedure of the proposed Diffusion Process is addressed as follows:

Step 1: Set maximum diffusion number (MDN points)

Step 2: Initialize *i* = 1.

Step 3: Compare i with MDN. If *i* is less than or equal MDN go to Step 5. Otherwise go to Step 4.

Step 4: Based on user define select Gaussian walk to create new position. Increase *i* by 1 and go to Step 3.

Step 5: Best point among created Gaussian walk is selected and go to main function.

3.2.3 The overall procedure of the proposed Updating Process is addressed as follows:

a. First updating process

Step 1: First, all points are ranked based on Eq. (16).

Step 2: For each point P_i in group.

Step 3: For each component *j* in P_i.

Step 4: If *Rank* $(P_i) \le$ rand[0,1] go to Step 5. Otherwise go to Step 3.

Step 5: Update jth component of P_i based on Eq.17.

b. Second updating process

Step 1: First, all points are ranked based on Eq. (16).

Step 2: For each point P_i in group

Step 3: If *Rank* $(P_i) \le$ rand[0,1] go to Step 4. Otherwise go to Step 2.

Step 4: Update position of P_i based on Eq.18 and Eq.19.

4. NUMERICAL RESULTS

This paper uses SFSA for solving the Economic dispatch (ED) problem in 3 cases

Case I: Microgrid system with 140 diesel generators and solved fuel constrained economic dispatch (ED) problem. This is ED problem considering valve-point effects prohibited operating zones and transmission losses [10]

Case II: Microgrid system with 40 generators with valve-point loading. This problem use formulas in ED problem considering valve-point effects for finding best results [15]

Case III: Microgrid system with two conventional generators (synchronous generators), one combined heat and power (CHP), wind generator, solar generator - in the islanded mode. This problem use formulas in ED problem considering prohibited operating zones and transmission losses and Economic Dispatch considering renewable energy cost functions [16]

The algorithm of this method was programmed by MATLAB R2010b in 1.3 GHz, core i5, personal computer

4.1. Case I

In this case, input parameter is taken from article [10] with total power need supply: P_D = 49342 (W) the results have been obtained to evaluate effectiveness of SFSA. In these systems, parameter a_i, b_i, c_i, e_i, f_i of generator were introduced in Tables 1 to 4.

Table 1: Parameter a_i

Unit *i*	a_i						
1-7	1220.65	1315.12	874.29	874.29	1976.47	1338.09	1818.3
8-14	1133.98	1320.64	1320.64	1320.64	1106.54	1176.50	1176.50
15-21	1176.504	1176.504	1017.406	1017.406	1229.131	1229.131	1229.131
22-38	1229.131	1267.894	1229.131	975.926	1532.093	641.989	641.989
29-35	911.533	910.533	1074.81	1074.81	1074.81	1074.81	1278.46
36-42	861.742	408.834	408.834	1288.815	1436.251	669.988	134.544
43-49	3427.912	3751.772	3918.78	3379.58	3345.296	3138.754	3453.05
50-56	5119.3	1898.415	1898.415	1898.415	1898.415	2473.39	2781.705
57-63	5515.508	3478.3	6240.909	9960.11	3671.997	1837.383	3108.395
64-70	3108.395	7095.484	3392.732	7095.484	7095.484	4288.32	13813.001
71-77	4435.493	9750.75	1042.366	1159.895	1159.895	1303.99	1156.193
78-84	2118.968	779.519	829.888	2333.69	2028.954	4412.017	2982.219
85-91	2982.219	3174.939	3218.359	3723.822	3551.405	4322.615	3493.739
92-98	226.799	382.932	156.987	154.484	332.834	326.599	345.306
99-105	350.372	370.377	367.067	124.875	130.785	878.746	827.959
106-112	432.007	445.606	467.223	475.94	899.462	1000.367	1269.132
113-119	1269.132	1269.132	4965.124	4965.124	4965.124	2243.185	2290.381
120-126	1681.533	6743.302	394.398	1243.165	1454.74	1011.051	909.269
127-133	689.378	1443.792	535.553	617.734	90.966	974.447	263.81
134-140	1335.594	1033.871	1391.325	4477.11	57.794	57.794	1258.437

Table 2: Parameter b_i

Unit *i*	b_i						
1-7	61.242	41.095	46.31	46.31	54.242	61.215	11.791
8-14	15.055	13.226	13.226	13.226	14.498	14.651	14.651
15-21	14.651	14.651	15.669	15.669	14.656	14.656	14.656
22-38	14.656	14.378	14.656	16.261	13.362	17.203	17.203
29-35	15.274	15.212	15.033	15.033	15.033	15.033	13.992

Table 2: Parameter b_i (continued)

Unit *i*	b_i						
36-42	15.679	16.542	16.542	16.518	15.815	75.464	129.544
43-49	56.613	54.451	54.736	58.034	55.981	61.52	58.635
50-56	44.647	71.584	71.584	71.584	71.584	85.12	87.682
57-63	69.532	78.339	58.172	46.636	76.947	80.761	70.136
64-70	70.136	49.84	65.404	49.84	49.84	66.465	22.941
71-77	64.314	45.017	70.644	70.959	70.959	70.302	70.662
78-84	71.101	37.854	37.768	67.983	77.838	63.671	79.458
85-91	79.458	93.966	94.723	66.919	68.185	60.821	68.551
92-98	2.842	2.946	3.096	3.04	1.709	1.668	1.789
99-105	1.815	2.726	2.732	2.651	2.798	1.595	1.503
106-112	2.425	2.499	2.674	2.692	1.633	1.816	89.83
113-119	89.83	89.83	64.125	64.125	64.125	76.129	81.805
120-126	81.14	46.665	78.412	112.088	90.871	97.116	83.244
127-133	95.665	91.202	104.501	83.015	127.795	77.929	92.779
134-140	80.95	89.073	161.288	161.829	84.972	84.972	16.087

Table 3: Parameter c_i

Unit *i*	c_i						
1-7	0.032888	0.00828	0.003849	0.003849	0.042468	0.014992	0.007039
8-14	0.003079	0.005063	0.005063	0.005063	0.003552	0.003901	0.003901
15-21	0.003901	0.003901	0.002393	0.002393	0.003684	0.003684	0.003684
22-38	0.003684	0.004004	0.003684	0.001619	0.005093	0.000993	0.000993
29-35	0.002473	0.002547	0.003542	0.003542	0.003542	0.003542	0.003132
36-42	0.001323	0.00295	0.00295	0.000991	0.001581	0.90236	0.110295
43-49	0.024493	0.029156	0.024667	0.016517	0.026584	0.00754	0.01643
50-56	0.045934	0.000044	0.000044	0.000044	0.000044	0.002528	0.000131
57-63	0.010372	0.007627	0.012464	0.039441	0.007278	0.000044	0.000044
64-70	0.000044	0.018827	0.010852	0.018827	0.018827	0.03456	0.08154
71-77	0.023534	0.035475	0.000915	0.000044	0.000044	0.001307	0.000392
78-84	0.000087	0.000521	0.000498	0.001046	0.13205	0.096968	0.054868
85-91	0.054868	0.014382	0.013161	0.016033	0.013653	0.028148	0.01347
92-98	0.000064	0.000252	0.000022	0.000022	0.000203	0.000198	0.000215
99-105	0.000218	0.000193	0.000197	0.000324	0.000344	0.00069	0.00065
106-112	0.000233	0.000239	0.000261	0.0002590	0.000707	0.000786	0.014355
113-119	0.014355	0.014355	0.030266	0.030266	0.030266	0.024027	0.00158
120-126	0.022095	0.07681	0.953443	0.000044	0.072468	0.000448	0.599112
127-133	0.244706	0.000042	0.085145	0.524718	0.176515	0.063414	2.740485
134-140	0.112438	0.041529	0.000911	0.005245	0.234787	0.234787	1.111878

Table 4: Parameters e_i and f_i

Unit *i*	e_i							f_i							
1-7	0	0	0	0	700	0	0	0	0	0	0	0.08	0	0	0
8-14	0	0	600	0	0	0	0	0	0	0.055	0	0	0	0	0
15-21	800	0	0	0	0	0	0	0.06	0	0	0	0	0	0	0.06
22-38	600	0	0	0	0	0	0	0.05	0	0	0	0	0	0	0.05
29-35	0	0	0	0	600	0	0	0	0	0	0	0.043	0	0	0
36-42	0	0	0	0	600	0	0	0	0	0	0	0.043	0	0	0
43-49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50-56	1100	0	0	0	0	0	0	0.043	0	0	0	0	0	0	0.043
57-63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
64-70	0	0	0	0	1200	0	1000	0	0	0	0	0.03	0	0.05	0
71-77	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78-84	0	0	0	0	1000	0	0	0	0	0	0	0.05	0	0	0
85-91	0	0	0	0	0	600	0	0	0	0	0	0	0.07	0	0
92-98	1200	0	0	0	0	0	0	0.043	0	0	0	0	0	0	0.043
99-105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
106-112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
113-119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120-126	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
127-133	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
134-140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The algorithm SFSA was tested with loop count N: 500, 5000 and 10000. Power of generators and total fuel cost recieved by the testing process were compared with extremophiles condition of P_i (Tables 5 and 6).

Table 5: Parameter P_i min

Unit *i*	P_i **min (W)**						
1-7	71	120	125	125	90	90	280
8-14	280	260	260	260	260	260	260
15-21	260	260	260	260	260	260	260
22-38	260	260	260	280	280	280	280
29-35	260	260	260	260	260	260	260
36-42	260	120	120	423	423	3	3
43-49	160	160	160	160	160	160	160
50-56	160	165	165	165	165	180	180
57-63	103	198	100	153	163	95	160
64-70	160	196	196	196	196	130	130
71-77	137	137	195	175	175	175	175
78-84	330	160	160	200	56	115	115
85-91	115	207	207	175	175	175	175
92-98	360	415	795	795	578	615	612
99-105	612	758	755	750	750	713	718
106-112	791	786	795	795	795	795	94
113-119	94	94	244	244	244	95	95
120-126	116	175	2	4	15	9	12
127-133	10	112	4	5	5	50	5
134-140	42	42	41	17	7	7	26

Table 6: Parameter P_i max

Unit *i*	P_i **max (W)**						
1-7	119	189	190	190	190	190	490
8-14	490	496	496	496	496	506	509
15-21	506	505	506	506	505	505	505
22-38	505	505	505	537	537	549	549

Unit *i*	P_i max (W)						
29-35	501	501	506	506	506	506	500
36-42	500	241	241	774	769	19	28
43-49	250	250	250	250	250	250	250
50-56	250	504	504	504	504	471	561
57-63	341	617	312	471	500	302	511
64-70	511	490	490	490	490	432	432
71-77	455	455	541	536	540	538	540
78-84	574	531	531	542	132	245	245
85-91	245	307	307	345	345	345	345
92-98	580	645	984	978	682	720	718
99-105	720	964	958	1007	1006	1013	1020
106-112	954	952	1006	1013	1021	1015	203
113-119	203	203	379	379	379	190	189
120-126	194	321	19	59	83	53	37
127-133	34	373	20	38	19	98	10
134-140	74	74	105	51	19	19	40

4.1.1 SFSA was tested with loop count 500

With 500 loop count, time of calculation (TC) is 85.801s, fuel cost (FC) is 1,564,639.51($). For this test case, generator powers still meet extremophiles condition of P_i. Table 7 compares with algorithm IPSO, CQGSO, DEL, IDE, and DE of article [10].

Table 7: Compare the best values of SFSA, IPSO, CQGSO, and DEL

Algorithm	FC	TC
SFSA	1,564,639.51	85.801
IPSO [10]	1,657,962.73	150
CQGSO [10]	1,657,962.72	31.67
DEL [10]	1,657,962.71	57.98
IDE [10]	1,564,648.66	27.88
DE [10]	1,566,264.99	27.84

Table 8: Generator distribution power with *N* = 500 of Case I

Unit *i*	P_i (W)						
1-7	108.2906	187.9000	190.0000	185.9163	90.0000	190.0000	490.0000
8-14	490.0000	496.0000	494.6695	496.0000	496.0000	506.0000	507.4191
15-21	506.0000	505.0000	506.0000	506.0000	505.0000	505.0000	505.0000
22-38	505.0000	505.0000	505.0000	537.0000	537.0000	549.0000	549.0000
29-35	501.0000	501.0000	506.0000	506.0000	506.0000	506.0000	500.0000
36-42	500.0000	241.0000	240.2901	774.0000	769.0000	5.4749	3.0000
43-49	160.0000	224.0731	229.6392	250.0000	250.0000	250.0000	250.0000
50-56	250.0000	165.0000	165.8045	165.0000	165.0000	182.1073	182.8138
57-63	103.0000	198.0000	312.0000	263.9845	163.0000	95.0000	160.0000
64-70	511.0000	490.0000	302.8765	490.0000	480.8811	130.0000	333.1933
71-77	137.0000	137.0000	195.0000	175.0000	175.0000	175.0000	175.0000
78-84	330.0000	529.5428	531.0000	200.0000	56.0000	115.0000	115.1548
85-91	115.0000	208.2919	207.0000	175.0000	175.0000	208.0021	175.0000
92-98	580.0000	644.4489	984.0000	978.0000	682.0000	720.0000	718.0000
99-105	720.0000	964.0000	958.0000	1,007.0000	1,006.0000	1,013.0000	1,020.0000
106-112	954.0000	951.9932	1,006.0000	1,011.9772	1,021.0000	1,015.0000	94.0000
113-119	94.0000	94.1961	244.0000	245.3610	244.0000	95.0000	95.0000
120-126	116.0000	175.0000	2.3257	4.8656	15.9959	9.0129	37.0000
127-133	10.0000	113.4207	4.0000	5.0000	5.0000	50.0000	5.1384
134-140	42.0135	42.0000	41.0000	17.0000	7.0000	8.9972	27.9225

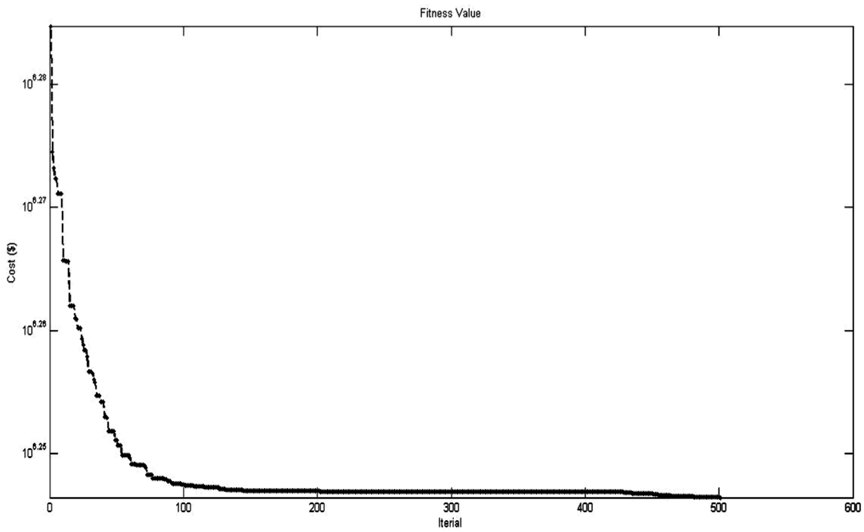

Fig. 1. Value of target function with *N*=500 of Case I.

The power distribution of 140 generators is represented by Table 8 and Figure 1 describes value of target function.

4.1.2 SFSA was tested with loop count 5000

With 5000 loop count, time of caculation (TC) is 659.02s, fuel cost (FC) is 1,564,612.86($). For this test case, generator powers still meet extremophiles condition of P_i. TC is approximately 8 times longer than TC at N=500. FC tends to decrease and approximate with the value of algorithms.

The power distribution of 140 generators is represented by Table 9 and Figure 2 describes value of target function.

Table 9: Generator distribution power with N=5000 of Case I

Unit *i*	***P_i (W)***						
1-7	107.1906	189.0000	190.0000	185.9163	90.0000	190.0000	490.0000
8-14	490.0000	496.0000	494.6695	496.0000	496.0000	506.0000	507.4191
15-21	506.0000	505.0000	506.0000	506.0000	505.0000	505.0000	505.0000
22-38	505.0000	505.0000	505.0000	537.0000	537.0000	549.0000	549.0000
29-35	501.0000	501.0000	506.0000	506.0000	506.0000	506.0000	500.0000
36-42	500.0000	241.0000	240.2899	774.0000	769.0000	5.4749	3.0000
43-49	160.0000	224.0734	229.6392	250.0000	250.0000	250.0000	250.0000
50-56	250.0000	165.0000	165.8045	165.0000	165.0000	182.1073	182.8138
57-63	103.0000	198.0000	312.0000	263.9845	163.0000	95.0000	160.0000
64-70	511.0000	490.0000	302.8765	490.0000	480.8811	130.0000	333.1933
71-77	137.0000	137.0000	195.0000	175.0000	175.0000	175.0000	175.0000
78-84	330.0000	529.5428	531.0000	200.0000	56.0000	115.0000	115.1548
85-91	115.0000	208.2919	207.0000	175.0000	175.0000	208.0020	175.0000
92-98	580.0000	644.4489	984.0000	978.0000	682.0000	720.0000	718.0000
99-105	720.0000	964.0000	958.0000	1,007.0000	1,006.0000	1,013.0000	1,020.0000
106-112	954.0000	951.9930	1,006.0000	1,011.9777	1,021.0000	1,015.0000	94.0000
113-119	94.0000	94.1954	244.0000	245.3623	244.0000	95.0000	95.0000
120-126	116.0000	175.0000	2.3257	4.8656	15.9959	9.0129	37.0000
127-133	10.0000	113.4207	4.0000	5.0000	5.0000	50.0000	5.1384
134-140	42.0168	42.0000	41.0000	17.0000	7.0000	8.9972	27.9225

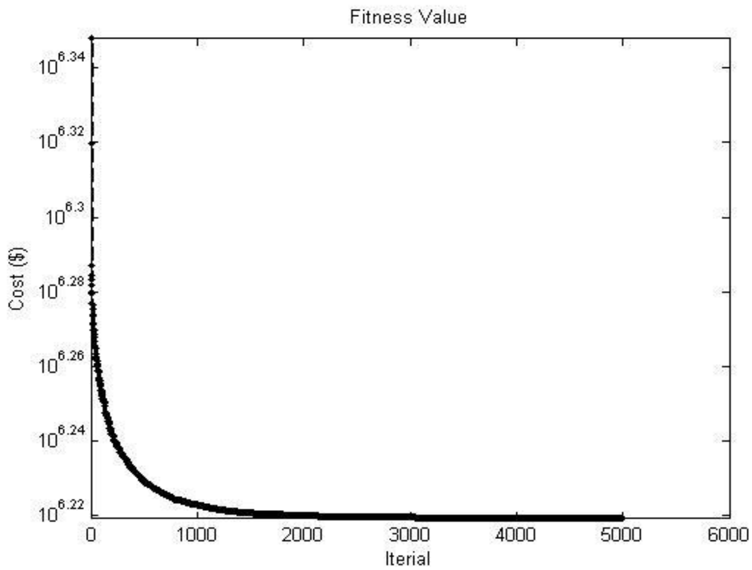

Fig. 2. Value of target function with N=5000 of Case I.

4.1.3 SFSA was tested with loop count 10000

With 10000 loop count, time of calculation (TC) is 1506.76s, fuel cost (FC) is 1,564,612.8594($). For this test case, generator powers still meet extremophiles condition of P_i. TC is approximately 3 times longer than TC at N = 5000. FC tends to decrease and approximate with the value of algorithms. The power distribution of 140 generators is represented by Table 10 and Figure 3 describes value of target function

Table 10: Generator distribution power with *N*=10000 of Case I

Unit *i*	***P_i (W)***						
1-7	107.1906	189.0000	190.0000	185.9163	90.0000	190.0000	490.0000
8-14	490.0000	496.0000	494.6695	496.0000	496.0000	506.0000	507.4191
15-21	506.0000	505.0000	506.0000	506.0000	505.0000	505.0000	505.0000
22-38	505.0000	505.0000	505.0000	537.0000	537.0000	549.0000	549.0000
29-35	501.0000	501.0000	506.0000	506.0000	506.0000	506.0000	500.0000
36-42	500.0000	241.0000	240.2899	774.0000	769.0000	5.4749	3.0000
43-49	160.0000	224.0734	229.6392	250.0000	250.0000	250.0000	250.0000
50-56	250.0000	165.0000	165.8045	165.0000	165.0000	182.1073	182.8138
57-63	103.0000	198.0000	312.0000	263.9845	163.0000	95.0000	160.0000
64-70	511.0000	490.0000	302.8765	490.0000	480.8811	130.0000	333.1933
71-77	137.0000	137.0000	195.0000	175.0000	175.0000	175.0000	175.0000
78-84	330.0000	529.5428	531.0000	200.0000	56.0000	115.0000	115.1548
85-91	115.0000	208.2919	207.0000	175.0000	175.0000	208.0020	175.0000
92-98	580.0000	644.4489	984.0000	978.0000	682.0000	720.0000	718.0000
99-105	720.0000	964.0000	958.0000	1,007.0000	1,006.0000	1,013.0000	1,020.0000
106-112	954.0000	951.9930	1,006.0000	1,011.9777	1,021.0000	1,015.0000	94.0000
113-119	94.0000	94.1954	244.0000	245.3623	244.0000	95.0000	95.0000
120-126	116.0000	175.0000	2.3257	4.8656	15.9959	9.0129	37.0000
127-133	10.0000	113.4207	4.0000	5.0000	5.0000	50.0000	5.1384
134-140	42.0168	42.0000	41.0000	17.0000	7.0000	8.9972	27.9225

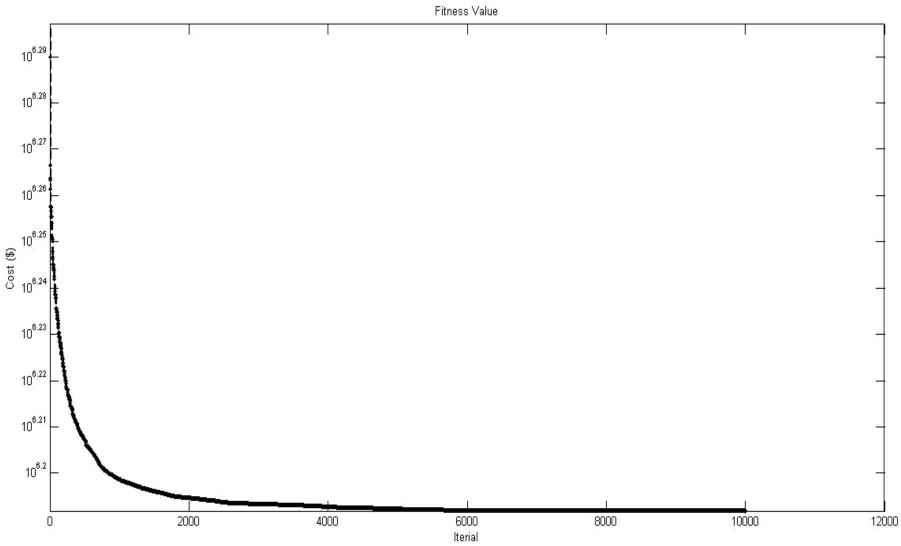

Fig. 3. Value of target function with N=10000 of Case I.

4.2. Case II

In this case, input parameter is taken from article [15] with total power need supply: PD = 10500 (MW) the results have been obtained to evaluate effectiveness of SFSA. In these systems, parameter a_i, b_i, c_i, e_j, f_j ,P_{imn}, P_{imax} of generators were introduced in Table 11.

Table 11: Unit data for 40 unit case of Case II

U-nit	P_{imin} (MW)	P_{imax} (MW)	a_i	b_i	c_i	e_i	f_i
1	36	114	0.00690	6.73	94.705	100	0.084
2	36	114	0.00690	6.73	94.705	100	0.084
3	60	120	0.02028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.60	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.20	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.40	300	0.035
14	125	500	0.00752	8.84	1750.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	646.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.10	801.32	300	0.035
26	254	550	0.00277	7.10	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.01140	5.35	148.89	120	0.077
31	60	190	0.00160	6.43	222.92	150	0.063
32	60	190	0.00160	6.43	222.92	150	0.063
33	60	190	0.00160	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

With 500 loop count, time of calculation (TC) is 7,176.31s, fuel cost (FC) is 121,473.97($). For this test case, generator powers still meet extremophiles condition of Pi. Table 12 compares with algorithms CEP, FEP, MFEP, and IFEP of article [15]. Figure 4 describes value of target function.

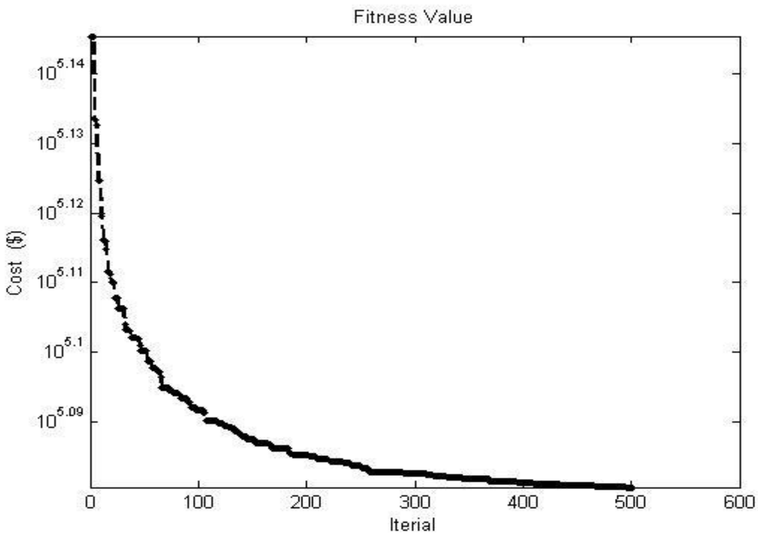

Fig. 4. Value of target function with N = 500 of Case II.

Table 12: Compare best values of SFSA, CEP, FEP, MFEP, and IFEP of Case II

Algorithm	FC	TC
SFSA	121,473.97	7,176.31
CEP [15]	123,488.29	1,955.2
FEP [15]	122,679.71	1,037.9
MFEP [15]	122,647.57	2,194.7
IFEP [15]	122,624.35	1,167.35

Table 13: Load demand for 24 hours

Time (hour)	Load (MW)	Time (hour)	Load (MW)
1	140	13	240
2	150	14	220
3	155	15	200
4	160	16	180
5	165	17	170
6	170	18	185
7	175	19	200
8	180	20	240
9	210	21	225
10	230	22	190
11	240	23	160
12	250	24	145

4.3. Case III

In this case, input parameters is taken from article [16] with total power need supply were introduced in Table 13. The results have been obtained to evaluate effectiveness of SFSA. In these systems, parameters a_i, b_i, c_i, and a, I^p, G^E of generators were introduced in Table 14. The power generation of wind turbines and PV were introduced in Table 15.

Table 14: Input parameter of generators

	CHP	**Gen 1**	**Gen 2**	**Solar**	**Wind**
c_i ($/h)	0.024	0.029	0.021	-	-
b_i ($/h)	21	20.16	20.4	-	-
a_i ($/h)	1530	992	600	-	-
a	-	-	-		
I^p($/kW)	-	-	-	5000	1400
G^E($/kW)	-	-	-	0.016	0.016

Table 15: Power generation of wind and solar for 24 hours

Time (hour)	Load (MW)		**Time (hour)**	Load (MW)	
	Wind	Solar		Wind	Solar
1	1.7	0	**13**	63.37	31.94
2	8.5	0	**14**	93.11	26.81
3	9.27	0	**15**	93.91	10.08
4	16.66	0	**16**	5.30	13.71
5	7.22	0	**17**	9.57	3.44
6	4.91	0.03	**18**	2.31	1.87
7	112.67	6.27	**19**	0.00	0.75
8	124.65	16.98	**20**	0.00	0.17
9	105.20	24.05	**21**	0.00	0.15
10	99.95	39.37	**22**	0.00	0.31
11	67.99	7.41	**23**	0.00	1.07
12	62.60	3.65	**24**	0.00	0.58

With 50 loop count, this test case, generator powers still meet extremophiles condition of P_i. Table 16 compares the obtained result to that from the article in [16].

Table 16: Compare the best values of SFSA and RGM

Time (hour)	SFSA	RGM [16]
1	6,135.18	6,297.09
2	6,294.58	6,473.77
3	6,401.05	6,564.89
4	6,440.90	6,650.21
5	6,651.78	6,759.40
6	6,793.04	6,866.64
7	7,138.55	7,209.32
8	7,702.59	7,761.62
9	8,824.90	8,648.51
10	10,130.85	9,712.91
11	8,749.62	8,721.92
12	8,729.48	8,793.72
13	10,012.76	9,653.81
14	9,314.96	9,013.44
15	7,987.90	7,904.86
16	7,212.42	7,268.33
17	7,313.69	7,276.08
18	7,295.32	7,288.47
19	7,540.49	7,543.59
20	8,510.25	8,567.35
21	8,146.25	8,167.21
22	7,308.39	7,314.42
23	6,599.78	6,674.43
24	6,260.59	6,388.52

5. CONCLUSION

This paper has proposed the SFSA method, as a new evolutionary technique, to solve economic dispatch problems. This paper's method employs two main processes including: Diffusion process and Update process. In this case 1 with loop count 500 SFSA made a good optimization result but the convergence time is slower than CQGSO, DEL. In this case 2 SFSA made a best optimization result and convergence time is 7 times slower than best TC of article [15]. In this case 3 SFSA made best optimization result all 24 hours. According figues, convergence speed of SFSA doesn't fast. Calculation time of SFSA at loop count 500 is as good as other algorithm, but when loop count increases, the calculation time increases fast. So that we need to choose value of N for have optimum result with calculation time as fast as possible.

REFERENCES

[1] Noel Augustine, Sindhu Suresh, Prajakta Moghe, Kashif Sheikh. 2012. Economic Dispatch for a Microgrid Considering Renewable Energy Cost Functions. *IEEE PES Innovative Smart Grid Technologies* (ISGT), 16-20 January. 2012, 1-7

[2] Seon-Ju Ahn, Soon-Ryul Nam, Joon-Ho Choi, Seung-Il Moon. 2013. Power Scheduling of Distributed Generators for Economic and Stable Operation of a Microgrid. *IEEE Transactions on Smart Grid*, 4(1): 398-405.

[3] Florian Dorfler, John Simpson-Porco , Francesco Bullo. 2014. Breaking the Hierarchy: Distributed Control & Economic Optimality in Microgrids. *IEEE Transactions on Control of Network Systems*, 99, 1.

[4] Anderson Hoke, Alexander Brissette, Shawn Chandler, Annabelle Pratt. 2013. Look-ahead economic dispatch of microgrids with energy storage, using linear programming. *IEEE Conference on Technologies for Sustainability* (SusTech), 2013, 154-161.

[5] Gwo-Ching Liao 2012. Solve environmental economic dispatch of Smart MicroGrid containing distributed generation system – Using chaotic quantum genetic algorithm. *International Journal of Electrical Power & Energy Systems*, 43(1): 779-787.

[6] Hongbin Wu, Xingyue Liu, Ming Ding. 2014. Dynamic economic dispatch of a microgrid: Mathematical models and solution algorithm. *International Journal of Electrical Power & Energy Systems*, 63, 336-346.

[7] M. Basu, A. Chowdhury. 2013. Cuckoo search algorithm for economic dispatch. International *Journal of Electrical Power & Energy Systems*, 60, 99-108

[8] Gwo-Ching Liao. 2013. The optimal economic dispatch of smart Microgrid including Distributed Generation. *2013 IEEE International Symposium on Next-Generation Electronics* (ISNE), 25-26 February 2013, 473-477

[9] Zwe-Lee Gaing. 2003. Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Transactions on Power Systems*, 18(3), 1187 – 1195

[10] M. Basu. 2014. Improved differential evolution for economic dispatch. *International Journal of Electrical Power & Energy Systems*, 63, 855-861

[11] Liu Xiaoping, Ding Ming, Han Jianghong, Han Pingping. 2010. Dynamic economic dispatch for microgrids including battery energy storage. *The 2nd International Symposium on Power Electronics for Distributed Generation Systems*, 914-917

[12] F. García, C. Bordons. 2013. Optimal economic dispatch for renewable energy microgrids with hybrid storage using Model Predictive Control. *Annual Conference of the IEEE on Industrial Electronics Society*, IECON 2013 - 39th, 10-13 November 2013, 7932-7937

[13] B.B. Mandelbrot.(1983). *The Fractal Geometry of Nature*. New York: W. H. Freeman and Co**.**

[14] Hamid Salimi. 2015. Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowledge-Based Systems***,** 75: 1-18

[15] Nidul Sinha, R. Chakrabarti, and P. K. Chattopadhyay. 2003. Evolutionary Programming Techniques for Economic Load Dispatch. *IEEE Transactions on evolutionary computer* 7(1): 83-94.

[16] Noel Augustine, Sindhu Suresh, Prajakta Moghe and Kashif Sheikh. 2011. Economic Dispatch for a Microgrid Considering Renewable Energy Cost Functions. *Innovative Smart Grid Technologies* (ISGT), IEEE PES 2012.

[17] Yang HT, Yang PC, Huang CL. 1996. Evolutionary programming based economic dispatch for units with nonsmooth fuel cost functions. *IEEE Trans Power System* 11(1): 112-118.