



The Chaotic Social Group Optimization for the Economic Dispatch Problem

Dinu Calin Secui^{1*}Gabriel Bendea¹Monica Liana Secui²Cristina Hora¹Codruta Bendea¹

¹*Department of Energy Engineering, University of Oradea, Oradea, Romania*

²*Department of Psychology, University of Oradea, Oradea, Romania*

* Corresponding author's Email: csecui@uoradea.ro

Abstract: Economic dispatch problem is found in the recent focus of specialists, solving this problem being based especially on metaheuristic algorithms. Social Group Optimization (SGO) is a new metaheuristic algorithm that models human behavior to solve a certain problem, and it has been successfully applied to optimize mathematical functions. This paper investigates the behavior of the SGO algorithm equipped with five chaotic maps (Logistic, Double, Iterative, Singer and Cat), applied to solve the economic dispatch problem with multiple fuel sources. The new algorithm, called Chaotic Social Group Optimization (CSGO), results from including the sequences generated by chaotic maps in the heuristics specific to the SGO algorithm. The performance of the CSGO chaotic algorithm is tested on four systems having various dimensions (10-units, 20-units, 30-units, and 40-units) and characteristics. The SGO and CSGO algorithms are compared with each other and with other algorithms. The obtained results indicate that CSGO algorithms equipped with any of the chaotic maps mentioned has a higher average performance than SGO algorithm. The application of CSGO, compared to SGO, leads to a reduction of the fuel cost between (0.6-6.7) \$/h for the analyzed systems.

Keywords: Economic load dispatch, Multi-fuel sources, Chaotic maps, Social group optimization.

1. Introduction

Economic dispatch (ED) is an important topic in the optimal operation of power systems. The main objective of the ED problem is the optimal planning of the generating units that operate in the power system, so that the total cost of fuel is minimal and a series of technical restrictions regarding the generating units and the power system are maintained. The fuel cost for a generating unit is represented by either a quadratic function or a quadratic function to which a sinusoidal term is added in order to take into account the valve-point effects. Considering these effects together with other practical characteristics (such as: the transmission losses, ramp rate limits or the use of multiple fuel sources), it brings to a non-linear, non-continuous, non-convex optimization model.

Solving the optimization models of the ED problem has been made using either classical

methods or meta-heuristic methods. The classical methods (such as: quadratic programming [1], non-linear programming [2] etc), based on the calculation of derivatives, encountered difficulties in identifying the global solution due to the nonlinearities in the optimization model. Implementing these methods is relatively difficult. Also, the results obtained are modest, considering the quality of the solutions. Successful application of traditional methods assumes that the cost-power function is monotonically increasing, a condition that reduces the accuracy of the calculations. Moreover, [3] states that the ED problem that takes into account practical issues such as ramp rate limits, valve-point effects, and multifuel options "is a challenging one and cannot be solved by the traditional methods".

To overcome these difficulties the ED problem has been solved using metaheuristic algorithms that are based on various processes inspired by biology, physics, chemistry, but also by social behavior of

living things. The main benefits of using metaheuristic algorithms compared to classical methods are [4]: handling constraints is relatively simple, metaheuristic algorithms are not sensitive to the nature of the objective function and to the range of values (convex/non-convex, discrete/continuous), continuity and derivability conditions are not imposed to the objective function or functions that define the constraints, metaheuristic algorithms do not use techniques that necessarily include the calculation of derivatives.

Some metaheuristic algorithms applied in their original form to solve the ED problem are mentioned below: differential evolution (DE) [4], particle swarm optimization (PSO) [5], artificial bee colony (ABC) optimization [6], artificial algae algorithm (AAA) [7], sine-cosine algorithm (SCA) [8], Coulomb's and Franklin's Laws Based Optimization (CFLBO) [9], slime mould algorithm (SMA) [10], squirrel search optimizer (SSO) [11] etc.

To increase the performance of the original metaheuristic algorithms, several strategies were used, such as: developing new varieties of original algorithms (by replacing the equations that update the solutions or by controlling some algorithm-specific parameters), creating hybrid algorithms, introducing chaos in the algorithms etc.

Some methods that use varieties or hybrids of the original algorithms to solve the ED problem are presented below. In [12] classical PSO algorithm is enhanced by linear decreasing of the multiple inertia weights (called MIW-PSO) to obtain a more efficient algorithm for solving the combined economic and emission load dispatch problem. The MIW-PSO is applied to the study of two small systems, and the results obtained indicate a significant reduction in cost and emissions. A novel dual-population adaptive differential evolution (DPADE) algorithm is presented in [13] for solving the ED problem taking into account the valve-point effects and multifuel sources. The DPADE uses a dual-population and adaptive technology to adjust the control parameters. It is applied to the study of large-scale systems considering up to 1280-units. To improve the solutions quality in the optimal power flow problem in [14] the authors proposed a modified ABC algorithm in which a percentage of the worst solutions were replaced with the best solutions. Modified ABC was tested on the IEEE 30 bus system, obtaining better results than other methods, such as: ABC, DE, GA, PSO. In [15] a clustering cuckoo search optimization (CCSO) is proposed, where the step size is determined by a clustering mechanism. The CCSO has been validated on

systems up to 40-units size and in all cases the CCSO has performed better than the original CSO.

A hybrid technique (HHE) for the multi-area economic dispatch problem is proposed in [16], efficiently combining the ability to explore and exploit of PSO and DE algorithms. In [17] a new hybrid is suggested that combines PSO with tabu search algorithm (TSA) and integrates the Sobol distribution to generate values for the inertia factor. The new DSPSO-TSA hybrid is more effective than well-known approaches, such as: GA, PSO, TSA or their varieties. The β -GWO [18] is a new hybrid that effectively integrates β -hill climbing optimizer (local search technique) with gray wolf optimizer (GWO) to solve the ED problem. A conglomerated optimizer is developed in [19] based on ion-motion (IMO) and crisscross search (CCS) algorithms. The resulting hybrid was tested on several functions (unimodal and multimodal) and on the ED problem.

Another commonly used method to improve the search capability of metaheuristic algorithms is to use chaotic maps. Some of the algorithms that adopt this strategy for optimizing mathematical functions or for solving ED problems are: chaotic sine cosine firefly algorithm [20], chaotic symbiotic organisms search (SOS) algorithm [21], chaotic JA with application in solving ED problem [22], chaotic SCA for economic emission dispatch problem [23], modified SOS algorithm with logistic map [24], etc. The chaotic maps are inserted in various phases of the optimization process, in the above-mentioned algorithms.

Relatively recently (2016) a new metaheuristic algorithm, called Social Group Optimization (SGO) [25], has been proposed for the optimization of standard numerical functions (taken from the IEEE Congress on Evolutionary Computation 2005 competition). The obtained results show that the SGO algorithm outperforms other well-known algorithms, such as [25, 26]: DE, PSO, ABC, GA (genetic algorithm), TLBO (teaching-learning-based optimization), harris hawks optimization (HHO), salp swarm algorithm (SSA), grasshopper optimization algorithm (GOA), socio evolution and learning optimization (SELO), volleyball premier league (VPL), SSA or its variants.

Based on the SGO algorithm, in this paper a new chaotic algorithm is proposed, called Chaotic SGO (CSGO). The new CSGO algorithm is obtained from the original SGO algorithm in which the random numbers are replaced with chaotic sequences generated by five chaotic maps (Logistic, Double, Iterative, Singer and Cat). The insertion of chaotic maps in the structure of the CSGO algorithm is motivated by the following arguments: (i). the SGO

algorithm is efficient in optimizing numerical functions with various dimensions and characteristics (unimodal, multimodal, composite) [25]. Thus, it is expected that in the case of ED problem this efficiency will be maintained; (ii). Consulting several databases (Scopus, Springer, Elsevier), we found that the SGO algorithm equipped with various chaotic maps has not been used in solving the ED problem; (iii). Also, CSGO algorithms are easy to implement and have the ability to maintain a good balance between exploration and exploitation, thus being able to generate promising solutions during the iterative process.

The main contributions of the research are:

(i). development of a chaotic optimization algorithm by combining the SGO algorithm with various chaotic maps (the new algorithm is called CSGO); (ii). implementation of CSGO algorithms to solve the multi-fuel source ED problem;

(iii). testing the performance of the CSGO algorithm on five power systems of different dimensions.

The paper is organized as follows: Section 2 presents the optimization model for the ED problem with valve-point effects and multiple fuel sources. Section 3 and 4 describes the SGO and CSGO algorithms. Section 5 details how to implement CSGO algorithms to solve the ED problem. Section 6 presents the analyzed cases and shows the obtained results. The conclusions are summarized in Section 7.

2. Formulation of ED problem

The ED problem aims to determine the powers of the generating units so that the total cost of fuel for the entire system is minimal while keeping the imposed restrictions. If we consider a power system with n units, the solution vector is represented as $[P]=[P_1, P_2, \dots, P_j, \dots, P_n]$, where P_j are the active powers of the generating unit j , in MW. Considering multiple fuel sources and the valve-point effects the cost-power characteristic $F_j(P_j)$ for a generating unit j is expressed as [27]:

$$F_j(P_j) = c_{jm}P_j^2 + b_{jm}P_j + a_{jm} + |e_{jm}\sin(f_{jm}(P_{min,jm} - P_j))| \quad (1)$$

if $P_{min,jm} \leq P_j \leq P_{max,jm}$

where, a_{jm} , b_{jm} , c_{jm} , e_{jm} , f_{jm} the fuel cost coefficients for unit d for a type m fuel; m is the type of fuel used for a unit j ; $P_{min,jm}$, $P_{max,jm}$ represent the minimum and maximum limits between which a unit j operates with a fuel of type m .

Characteristic $F_j(P_j)$ is composed of a quadratic component (having coefficients a_{jm} , b_{jm} and c_{jm}) and

a sinusoidal component reflecting the valve-point effects (with coefficients e_{jm} and f_{jm}).

The objective function is expressed by the total cost of the fuel used at the level of the entire power system, as follows [3]:

$$\min F[P] = \sum_{j=1}^n F_j(P_j) \quad (2)$$

The constraints of the ED problem are represented by a set of equations and inequalities presented below [3]:

$$P_{j,min} \leq P_j \leq P_{j,max}, j=1,2,\dots,n \quad (3)$$

$$P_j \leq P_j^0 + UR_j, \text{ if active power increases} \quad (4)$$

$$P_j \geq P_j^0 - DR_j, \text{ if active power decreases} \quad (5)$$

$$P_G - P_L - P_D = 0 \quad (6)$$

where $P_{j,min}$ and $P_{j,max}$ are the minimum and maximum operating limits of a generator j ; where P_j^0 is the previous hour active power of unit j ; DR_j and UR_j are the down-ramp and up-ramp limits of the j unit; P_D is the load demand; P_L is the transmission loss; P_G is the power generated in the system. The transmission losses P_L are calculated using constant B coefficient formula:

$$P_L = \sum_{i=1}^n \sum_{j=1}^n P_i B_{ij} P_j + \sum_{j=1}^n B_{0j} P_j + B_{00} \quad (7)$$

where B_{ij} is an element of the loss coefficient matrix, B_{0j} is j element of the loss coefficient vector and B_{00} is the loss coefficient constant.

Relations Eq. (3) show the operating limits of unit j . Constraints Eqs. (4) and (5) define the operating limits for a unit j when the power from the previous-hour P_j^0 is known. The interval in which a unit j can operate is given by the down-ramp (DR_j) and up-ramp (UR_j) limits, which show how much it can increase, respectively decrease the power of the unit j compared to the value P_j^0 . The equality constraint Eq. (6) defines the active power balance for the whole system, which shows that the total power generated (P_G) is equal to the required power (P_D) plus the transmission losses (P_L).

3. Social group optimization (SGO)

SGO is a population-based optimization method that mimics people's social behavior to solve a complex real-life problem. In SGO the population is represented by a group of people, who try solving a given problem based on the level of knowledge acquired. Each person is associated with a candidate

solution. The fitness function is associated with a person's ability to solve a given problem. For the implementation of the artificial SGO algorithm we use the following mathematical notations: $X_i=[x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{in}]_{i=1,2,\dots,N}$ is a solution vector that includes a number of traits of a person from the social group; x_{ij} is the component j of the solution vector X_i ; $f_i(X_i)=f_i$ is the fitness function corresponding to solution X_i ; n is the number of person traits, and N is the number of persons (solutions) from the social group.

SGO algorithm has four main phases: initialization phase, improving phase, acquiring phase, termination phase. In the initialization phase the solutions are randomly generated between the minimum and maximum limits, using the relation [25]:

$$x_{i,j}=x_{min,j}+r_1 \cdot (x_{max,j} - x_{min,j}), i=1, 2, \dots, N \quad (8)$$

where, $x_{min,j}$, $x_{max,j}$, $j=1, 2, \dots, n$ are the minimum and maximum limits of component j associated with the solution X_i ; r_1 is a random number uniformly distributed between 0 and 1.

In the improving phase the best person (the best solution) from the group tries spreading the knowledge among the other members to improve the level of knowledge in the social group. Updating the knowledge of a person from the social group is done using Algorithm 1 [25]:

Algorithm 1: Improving phase

For $i=1$ to N **do**

For $j=1$ to n **do**

$$x_{new,ij}=c \cdot x_{ij} + r_2 \cdot (Gbest_j - x_{ij}) \quad (9)$$

End; End;

The $X_{new,i}$ solution is accepted if $f(X_{new,i})$ is better than $f(X_i)$.

where, $X_{new,i}$ is the new solution vector obtained after the application of Algorithm 1; $x_{new,ij}$ is the component j of the vector $X_{new,i}$; X_i is the old solution vector, before the application of algorithm 1; x_{ij} is the component j of the vector X_i ; $Gbest_j$ is the best solution obtained so far; $Gbest_j$ is the component j of the $Gbest$ solution; c is the self-introspection parameter with values between 0 and 1; r_2 is a uniformly generated random number in the range (0, 1);

In the acquiring phase, each person from the social group interacts with the best person, as well as with other group members, to gain new knowledge. Updating the knowledge of a person from the social group is done using Algorithm 2 [25]:

Algorithm 2: Acquiring phase

For $i=1$ to N **do**

Randomly select a solution X_r , with $i \neq r$

If $f(X_i) < f(X_r)$ **Then**

For $j=1$ to n **do**

$$x_{new,ij}=x_{ij}+r_3(x_{ij}-x_{ij})+r_4(Gbest_j-x_{ij}) \quad (11)$$

End For

Else

For $j=1$ to n **do**

$$x_{new,ij}=x_{ij}+r_3(x_{ij}-x_{ij})+r_4(Gbest_j-x_{ij}) \quad (12)$$

End For

End If

The $X_{new,i}$ solution is accepted if $f(X_{new,i})$ is better than $f(X_i)$;

End For.

where, X_r , $r \neq i$ is a solution vector randomly selected from the current population; r_3 and r_4 are uniformly generated random numbers in the range (0, 1);

The last phase refers to the termination criterion of the algorithm. Stopping the algorithm is performed if the maximum number of k_{max} iterations is reached.

4. Chaotic SGO (CSGO)

The insertion of chaotic maps in the structure of metaheuristic algorithms can increase the efficiency of the new algorithm [22, 28]. Equipping the SGO algorithm with chaotic maps aims to improve the capacity of the CSGO algorithm to avoid local minimums, to increase the stability and strengthen the global search. In this paper, the SGO algorithm is combined with five chaotic maps: Logistic, Iterative, Double, Singer and Cat. The characteristics of the chaotic maps mentioned are presented below:

1. Logistic map [22]: $x_{k+1}=a \cdot x_k(1-x_k)$, $a=4$, $x_k \in (0,1)$;
2. Iterative map [28]: $x_{k+1}=\sin(a\pi/x_k)$, $a=2$, $x_k \in [-1,1] \setminus \{0\}$;
3. Double map [29]: $x_{k+1}=\sin(2a\pi x_k)$, $a=2$, $x_k \in (-1,1)$;
4. Singer map [28]: $x_{k+1}=a \cdot (7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.3x_k^4)$, $a=1.06$, $x_k \in (0.077, 0.987)$;
5. Cat map [30]: $x_{k+1} = (x_k + ay_k) \bmod 1$;
 $y_{k+1} = (bx_k + (ab+1)y_k) \bmod 1$; $x_k, y_k \in (0,1)$, $a=b=1$.

The CSGO algorithm has the same structure and the same steps as the original SGO algorithm [25], but the random numbers from the initialization phase (r_1), improving phase (r_2) and acquiring phase (r_3 and r_4) are replaced with sequences generated by the mentioned chaotic maps (similar to

[28]). Basically, using the five chaotic maps, we get five chaotic optimization algorithms, their name abbreviation being “CSGO-chaotic map name”. In order to compare the chaotic maps in the closest possible conditions, the initial values cx_0 used for all the cases studied were chosen as equal to $cx_0=0.1777$. In the initialization phase the chaotic sequences will be generated in the range (0, 1), and in the improving phase and acquiring phase in the range (-1, 1).

5. Implementation of CSGO for ED problem

The CSGO algorithms applied to solve the ED problem with multiple fuel sources include the following steps:

Step 1: *Specifying input data for the ED problem:* number of units (n), cost coefficients (a, b, c, e, f); loss coefficients B_{ij}, B_{0i}, B_{00} ; load demand (P_D);

Step 2: Set the CSGO parameters: N and k_{max} ;

Step 3: *Initialization of solutions*

3.1 Initialization of iterations: $k=0$;

3.2 The chaotic sequences cx_1 are generated (cx_1 replace the random numbers r_1 in relation (8)), using one of the mentioned maps;

3.3 The initial solutions are randomly generated using relation (8), so that the constraints imposed on the ED problem are met;

3.4 Evaluation of the initial solutions ($P_i, i=1,2,\dots, N$); determination of the best initial solution and its cost;

For $k=1$ **Do** k_{max}

Step 4: *Update solutions P_i in the improving phase*

For $i=1$ **To** N **Do**

For $j=1$ **to** n **Do**

4.1 Update the chaotic sequences cx_2 , which replace random numbers r_2 in the relation (9);

4.2 Update the P_{ji} component of the solution P_i with relation (9);

4.3 The constraints of inequality (3)-(5) are verified: if the $P_{new,ji}$ power is outside the limits, then the mechanism for handling the inequality restrictions presented in [24] is applied; **End For** j ;

4.4 The equality constraint (6) is verified: if the new solution $P_{new,i}$ needs to be adjusted, then the mechanism for handling the equality restriction presented in [24] is applied; **End For** i ;

4.5 Evaluate the new solutions P_i , by relation (2);

4.6 Compare the new solution $P_{new,i}$ with the old solution P_i . If the new solution $P_{new,i}$ is better, then replace P_i with $P_{new,i}$;

4.7 Identify and retain the best solution G_{best} ;

Step 5: *Update solutions P_i in the acquiring phase*

For $i=1$ **To** N **Do**

5.1 Randomly select a solution $P_r, r \in \{1, 2, \dots, N\}, r \neq i$;

If $F(X_i) < F(X_r)$ **Then**

For $j=1$ **to** n **Do**

5.1 Update the chaotic sequences cx_3 and cx_4 , which replace random numbers r_3 and r_4 in the relation (11);

5.2 Update the P_{ji} component of the solution P_i with relation (11);

5.3 The constraints of inequality (3)-(5) are verified, similar to point 4.3; **End For** j ;

Else

For $j=1$ **to** n **Do**

5.4 Update the chaotic sequences cx_3 and cx_4 , which replace random numbers r_3 and r_4 in the relation (12);

5.5 Update the P_{ji} component of the solution P_i with relation (12);

5.6 The constraints of inequality (3)-(5) are verified, similar to point 4.3; **End For** j ; **End If**;

5.7 The equality constraint (6) is verified, similar to point 4.4; **End For** i ;

5.8 Evaluate the new solutions P_i , by relation (2);

5.9 Compare the new solution $P_{new,i}$ with the old solution P_i . If the new solution $P_{new,i}$ is better, then replace P_i with $P_{new,i}$;

5.10 Identify and retain the best solution G_{best} ;

Step 6: *Stopping the process:* the calculation process is stopped when the maximum number of iterations (k_{max}) is reached. {**End For** k }

6. Simulation results and comparison

The efficiency of the SGO algorithm and of the CSGO algorithms equipped with different chaotic maps is tested by analyzing four systems having 10-units, 20-units, 30-units, and 40-units.

Description of the analyzed systems: Depending on the characteristics of the systems, five cases are studied (C1-C5), presented below:

Case 1 (C1): A 10-units system with multiple fuel sources and valve-point loading effects is studied, the transmission losses being neglected. The cost coefficients a, b, c, e, f and the power limits of the units are taken from [27]. The load demand P_D is 2700 MW.

Case 2 (C2): The second case studies the 10-units system from case C1 taking into account the transmission losses in the power system. The loss coefficients B are considered from [36].

Case 3 (C3), case 4 (C4) and case 5 (C5): In these cases, the 20-units, 30-units and 40-units systems obtained by replicating twice, three times, respectively four times the 10-units system in case C1 are analyzed. The powers required in these systems are 5400 MW, 8100 MW, respectively 10800 MW.

For each case, the quality and robustness of the solutions is evaluated by 50 runs. Based on the performed runs, the following statistics were retained: best total fuel cost F (*Best cost F*), mean total fuel cost F (*Mean cost F*), worst total fuel cost

F (*Worst cost F*) and standard deviation (*SD*). All simulations were done in MathCAD, on a computer with an Intel i5 processor, having the characteristics: 2.2 GHz CPU and 4 GB of RAM.

Table 1. The values of the statistical items obtained by different algorithms (case C1-C5)

Algorithm	Case	Best cost F (\$/h)	Mean cost F (\$/h)	Worst cost F (\$/h)	SD (\$/h)	Rank
JA [31]	Case C1: 10-units	624.6819	626.1531	637.5108	1.658	-
CGA-MU [27]		624.7193	627.6087	633.8652	-	-
IGA-MU [27]		624.5178	625.8692	630.8705	-	-
DE [4]		624.4606	624.4724	624.4918	0.007	-
PSO-GM [32]		624.305	624.6749	625.0854	0.158	-
PSO-LRS [3]		624.2297	625.7887	628.3214	-	-
PSO [4]		624.2449	624.2543	624.2744	0.002	-
GC-Jaya [33]		624.2249	624.6812	628.28.11	0.660	-
CPSO [32]		624.1715	624.5493	624.7844	0.127	-
MSFLA [34]		624.11569	624.8958	628.3428	-	-
M_DE [16]		624.1026	-	-	-	-
HHE [16]		624.0671	-	-	-	-
CBPSO-RVM [32]		623.9588	624.0816	624.293	0.057	-
TSA [17]		624.3078	624.8285	635.0623	1.159	-
GA [17]		624.5050	624.7419	624.8169	0.100	-
SGO		623.9179	624.8831	635.6641	1.877	6
SGO-Logistic		623.9125	624.0583	624.1754	0.059	2
SGO-Double		623.9625	624.1068	624.2511	0.064	3
SGO-Iterative		623.9170	624.0363	624.2044	0.064	1
SGO-Singer		623.8491	624.2499	626.5394	0.762	5
SGO-Cat	623.9496	624.1073	624.2115	0.056	4	
SGO	Case C2: 10-units, with transmission losses	700.3177	701.2902	706.0056	1.849	6
SGO-Logistic		700.3155	700.4358	700.5558	0.053	3
SGO-Double		700.3809	700.4873	700.5795	0.048	4
SGO-Iterative		700.3486	700.4343	700.5176	0.041	2
SGO-Singer		700.2501	700.4154	700.9802	0.148	1
SGO-Cat		700.3553	700.4899	700.6026	0.050	5
SGO	Case C3: 20-units	1248.1867	1251.2317	1260.7457	3.331	6
SGO-Logistic		1248.4230	1248.9334	1249.3402	0.206	2
SGO-Double		1248.8628	1249.6126	1250.6042	0.465	5
SGO-Iterative		1248.4458	1248.9108	1249.2710	0.180	1
SGO-Singer		1247.8797	1249.3174	1253.2771	1.523	4
SGO-Cat		1248.6073	1249.0555	1249.4559	0.205	3
SGO	Case C4: 30-units	1872.2854	1877.8060	1888.2968	4.350	6
SGO-Logistic		1873.1560	1873.9909	1875.7981	0.420	1
SGO-Double		1874.0164	1875.6813	1877.8939	0.869	5
SGO-Iterative		1873.2144	1874.0914	1875.4884	0.470	3
SGO-Singer		1871.7917	1873.9710	1883.7271	2.299	2
SGO-Cat		1873.6577	1874.3583	1875.3154	0.354	4
IMO [19]	Case C5: 40-units	2537.2500	2600.5920	2669.6190	27.882	-
MIMO [19]		2496.8590	2503.5270	2520.2890	5.659	-
CSO [35]		2495.7880	2496.6341	2497.1320	0.248	-
SGO		2498.2374	2504.3423	2513.3058	4.618	6
SGO-Logistic		2496.9975	2497.5560	2497.9921	0.243	1
SGO-Double		2500.5846	2503.1080	2505.8080	1.342	5
SGO-Iterative		2498.5429	2499.7372	2501.2157	0.670	3
SGO-Singer		2495.8071	2498.4940	2508.4666	2.733	2
SGO-Cat		2499.0007	2500.3746	2503.4240	0.920	4

Table 2. The best solutions obtained by CSGO-Singer algorithms

P_i (MW)	Fuel*	Case C1	Case C2	Case C3	P_i (MW)	Case C5	
P_1	2	217.56666	228.93014	217.59196	P_1/P_{21}	217.19361	216.52372
P_2	1	209.97877	219.13867	210.47605	P_2/P_{22}	211.46316	213.45369
P_3	1	279.64900	301.82946	285.71802	P_3/P_{23}	278.61969	281.76781
P_4	3	240.31140	245.41738	239.37055	P_4/P_{24}	238.56236	240.17645
P_5	1	279.93518	306.00853	278.72821	P_5/P_{25}	282.47694	276.75469
P_6	3	239.10192	244.20803	237.89231	P_6/P_{26}	240.44619	237.08653
P_7	1	290.09924	316.18613	287.90651	P_7/P_{27}	287.51709	290.49508
P_8	3	240.04251	244.20805	242.05903	P_8/P_{28}	238.42965	238.96786
P_9	3	427.42777	439.99996	427.50543	P_9/P_{30}	424.65805	424.90014
P_{10}	1	275.88754	297.39817	274.61603	P_{10}/P_{31}	279.11293	278.78224
P_{11}	2	-	-	217.46760	P_{11}/P_{32}	216.60360	219.53770
P_{12}	1	-	-	213.19991	P_{12}/P_{33}	212.70350	212.44934
P_{13}	1	-	-	280.63986	P_{13}/P_{34}	278.75342	279.65979
P_{14}	3	-	-	238.02741	P_{14}/P_{35}	240.04397	238.29727
P_{15}	1	-	-	276.72947	P_{15}/P_{36}	280.16658	280.17394
P_{16}	3	-	-	238.69965	P_{16}/P_{37}	238.96720	240.44593
P_{17}	1	-	-	290.05387	P_{17}/P_{38}	294.56060	289.91067
P_{18}	3	-	-	241.52034	P_{18}/P_{39}	237.75849	237.35520
P_{19}	3	-	-	426.39275	P_{19}/P_{40}	427.30883	430.74312
P_{20}	1	-	-	275.40504	P_{20}/P_{21}	271.51107	275.66189
A	-	-5.457E-12	3.233E-06	-1.000E-11	A	-1.637E-11	
P_L	-	-	143.32451	-	-	-	-

The accuracy (A) is calculated using the relation: $A = P_G - P_L - P_D$; *The same type of fuel was considered for all cases (C1-C5)

Parameters setting: The parameters of the SGO and CSGO algorithms are the number of persons in the social group (N), the maximum number of iterations (k_{max}) and self-introspection parameter (c). In the case of SGO and CSGO algorithms, the parameter c was considered to have the value $c=1$, and the parameters N and k_{max} were set by experimental trials. The parameters N and k_{max} were selected based on the *Mean cost F* item, having the following values: 25 and 100 (for case C1), 30 and 100 (case C2), 35 and 100 (case C3), 50 and 150 (case C4), 60 and 200 (case C5).

The presentation of the best solutions: The statistics resulting from the application of SGO and CSGO algorithms, referring to the following items (*Best cost F*, *Mean cost F*, *Worst cost F* and *SD*) are shown in Table 1, for each analyzed case. It is noticed in Table 1 that the best cost (*Best cost F*) is obtained by CSGO-Singer algorithm, therefore, the best solutions obtained by applying this algorithm are presented in Table 2. For cases without power losses, the accuracy ($|A|$) of calculations is below 1E-10 MW, while for case having power losses (C2) the accuracy is below 1E-5 MW.

Comparison of the SGO and CSGO algorithms: Based on the statistical items resulting from the application of the SGO and CSGO algorithms (presented in Table 1), the following can be found:

(i). for all case studies, CSGO algorithms equipped with various chaotic maps (Logistic, Double, Iterative, Singer and Cat) have a higher mean performance (measured by the *Mean cost F* item) than the original SGO algorithm. Also, when considering the items *Worst cost F* and *SD*, the CSGO algorithms (CSGO-Logistic, CSGO-Double, CSGO-Iterative, CSGO-Singer and CSGO-Cat) obtain better values than SGO algorithm, for all cases;

(ii). CSGO algorithms have a very good stability (*SD* item has values below 1 \$/h) for all cases and for the majority of maps used. Exceptions are the CSGO-Singer for cases C3-C5 and the CSGO-Double for case C5; CSGO algorithms have better stability than the SGO algorithm for all cases and maps used;

(iii). based on the *Best cost F* item, for each case, several chaotic algorithms obtain better solutions than SGO: CSGO-Logistic (for cases C1, C2, and C5), CSGO-Iterative (for case C1) and CSGO-Singer (for all cases C1-C5);

(iv). CSGO algorithms maintain their stability when the size of the systems is increased from 10-units to 40-units;

(v). there are several chaotic algorithms that outperform the original SGO algorithm, taking into account all the items (*Best cost F*, *Mean cost F*, *Worst cost F* and *SD*): CSGO-Logistic (for cases C1, C2, and C5), CSGO-Iterative (for case C1) and CSGO-Singer (for all cases C1-C5).

Table 3. Average ranks and average rank difference (ARD)

Algorithm	AR	ARD between CSGO & SGO
SGO	6	-
CSGO-Logistic	1.8	4.2
CSGO-Double	4.4	1.6
CSGO-Iterative	2	4
CSGO-Singer	2.8	3.2
CSGO-Cat	4	2
CD value (Bonferroni-Dunn), at $\alpha=95\%$: 3.048		

Statistical testing: To verify whether there are statistically significant differences between the performance of SGO and the group of CSGO algorithms, Friedman test was conducted. The algorithms are ranked for each data set (corresponding to each case), the best performing algorithm (using the results for *Mean cost F* item) getting the rank of 1, the second best rank 2, and so on (Table 1). The Friedman test compares the average ranks of algorithms. The null-hypothesis states that all algorithms are equivalent, and so their average ranks should be equal. Table 3 shows the average ranks (AR) of all algorithms with respect to their performance.

The computed Friedman test statistic value is 11.352, being higher than 2.71, which represents the critical Friedman statistics value for a confidence level of $\alpha=95\%$. So, the null-hypothesis is rejected, indicating that the results obtained by the algorithms used in this study are significantly different.

In order to compare the performance of SGO with the performance of each CSGO algorithms, a pairwise post-hoc Bonferroni-Dunn test was used. The performance of two compared algorithms is significantly different if the corresponding average ranks differ by at least the critical difference (CD) value. The average rank difference (ARD) between each CSGO algorithm and SGO algorithm, and the CD value at 95% confidence level are presented in Table 3. We can notice that the ARD of CSGO-Logistic, CSGO-Iterative, and CSGO-Singer with respect to SGO algorithm is greater than the CD value (3.084). In the case of CSGO-Double and CSGO-Cat algorithm the difference is smaller than this critical value. Based on the statistic tests conducted, we can highlight that CSGO-Logistic, CSGO-Iterative, and respectively CSGO-Singer algorithm has a better mean performance as compared to SGO algorithm. CSGO-Double and CSGO-Cat algorithms do not differ from SGO algorithm in terms of performance obtained for *Mean cost F* item.

Comparison of SGO and CSGO to other algorithms: In the cases C1 (10-units) and C5 (40-units) several algorithms applied to solve the ED problem with multiple fuel sources were found in the

literature. Details regarding the competing algorithms (inspiration, mathematical modeling, implementation, parameter setting, etc.) are given in the references indicated in Table 1. Below is a short description of the competing algorithms.

A first group of algorithms refers to PSO [4] and several varieties: CPSO (advanced PSO), CBPSO-RVM (PSO with inertia weight factor, constriction factor, and real-valued mutation), PSO-GM (PSO with Gaussian mutation) [32], and PSO-LRS (PSO with local random search) [3]. PSO algorithms are population-based algorithms inspired by swarm intelligence. In classical PSO the particles movement is based on three components: inertial, cognitive, and social. PSOs are fast convergence algorithms, easy to implement, but suffer from premature convergence in the case of multi-modal functions (as is the case of ED problem). CPSO [32] introduces a constriction factor that adjusts the velocity of the particles. CBPSO-RVM, PSO-GM and PSO-LRS are PSO varieties that enhance the global or local search capability of PSO.

The second group consists of the GA algorithm [17] and its varieties (IGA-MU (improved genetic algorithm with multiplier updating (MU)), CGA-MU (conventional GA with MU)) [27]. The GA is a population-based global search algorithm that mimics the natural genetic process. GA is based on three operators: crossover, mutation, and selection. GAs have the ability to identify a global solution, but have a long computational time. IGA-MU and CGA-MU are variants of GA that improve the search efficiency in solutions space (SS) for the ED problem with multiple local optima. The tabu search algorithm (TSA) [17] is a local heuristic search method that exploits the vicinity of a given solution to obtain an improved solution.

The third group consists of DE algorithm and its varieties. The DE [4] is a population-based algorithm that uses operators similar to GA. DE has a rapid convergence, which can sometimes be premature towards a local solution. The M-DE is an improved variety of DE in which an additional momentum operation is introduced to maintain the quality of the SS search. The HHE (hybrid hierarchical evolution) is a hybrid that combines the exploration and exploitation capabilities of the PSO and DE algorithms to improve the search efficiency.

Another group is represented by the Jaya algorithms. The Jaya algorithm (JA) [31] is an easy-to-implement metaheuristic technique that has only two parameters: the maximum number of iterations and population size, being used to solve optimization problems with/without constraints. The JA has a fast convergence, but the solutions can

stagnate a number of iterations. To cover this shortcoming GC-Jaya [33] introduces a mutation operator to increase diversity and avoid stagnation at certain points.

The last group includes IMO and MIMO algorithms. Ion motion optimization (IMO) [19] is a population-based algorithm that models the phenomenon of ions attraction and repulsion. IMO consists of two phases: the liquid phase (which performs SS exploration) and the solid phase (which performs SS exploitation). The MIMO [19] is a modified IMO, which takes into account the repulsion between ions of the same type and the attraction between ions of different types in the liquid phase.

Thus, the SGO and CSGO algorithms will be compared to other algorithms, using the items *Best cost F*, *Mean cost F*, *Worst cost F*, and *SD*. For the other cases (C2-C4) there were no algorithms identified for solving the ED problem.

For case C1, based on the results from Table 1, the following can be noticed:

(i). the SGO and CSGO algorithms obtain better solutions than all the algorithms mentioned in Table 1 (such as: GA [17], PSO [4], DE [4], TSA [17], Jaya [31] or their varieties [3, 16, 27, 32-33], in terms of the *Best cost F* item;

(ii). CSGO algorithms equipped with Logistic and Iterative maps have a better performance than all the algorithms mentioned in Table 1, in terms of the *Best cost F*, *Mean cost F*, *Worst cost F* items.

In the case of C5, based on the results from Table 1, the following can be highlight:

(i). the performance of SGO algorithm is higher compared to IMO algorithm [19] in terms of all statistical items;

(ii). all CSGO algorithms have a better average performance than IMO and MIMO [19]; Also, CSGOs get a better value for *Worst cost F* and *SD* items than IMO and MIMO;

(iii). CSO algorithm has the best performance [35].

Convergence: In Fig. 1 two convergence characteristics are drawn for each case, one obtained by the SGO algorithm, and the other by the CSGO algorithm equipped with the best chaotic map (Singer map). The characteristics are drawn for the best solution obtained.

Cost values differ significantly from case to case. Therefore, for a more compact representation these costs are normalized in the range [0, 1]. For a better visibility of the convergence process, the graphs corresponding to cases C1-C3 are shifted with the value 0.01, and for cases C4-C5 with the value 0.02.

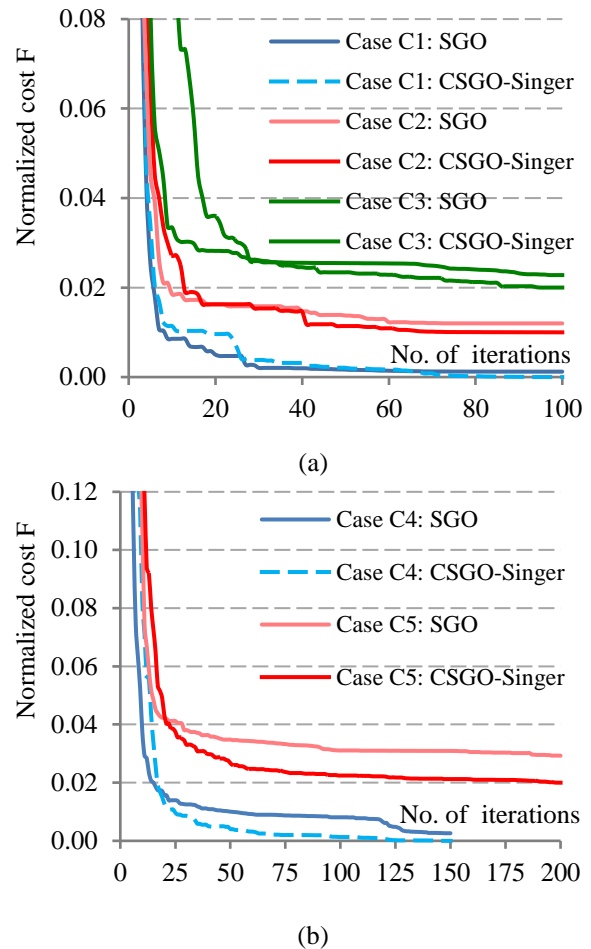


Figure. 1 Convergence characteristics obtained by SGO and CSGO-Singer for: (a) C1-C3 cases and (b) C4-C5 cases

Following the iterative process displayed in Fig. 1 it can be noticed that the convergence of the SGO and CSGO algorithms is good for all analyzed cases. Thus, after about 60-85% of the maximum number of iterations, the SGO and CSGO algorithms are able to obtain the best solution or a value close to it. Also, for each case, it can be seen that in the first 25-30 iterations the SGO algorithm has a faster convergence than CSGO-Singer, but after 30 iterations CSGO-Singer outperforms SGO, orienting toward the best solution (mentioned in Table 1).

7. Conclusions

This paper developed a chaotic SGO algorithm equipped with various chaotic maps that has been used to solve the economic dispatch problem with valve-point effects and multi-fuel sources. The chaotic sequences generated by various maps are included in the heuristics for updating the solutions contained in the initialization, improving and acquiring phases of SGO algorithm, increasing the

quality of the solutions and the stability of the new CSGO algorithm.

CSGO algorithms were applied on four small (10-units), medium (20-units and 30-units) and large (40-units) systems. In all analyzed cases (C1-C5), the obtained results indicate that the inclusion of chaotic maps in the structure of the SGO algorithm leads to increased stability (measured by *SD* item) and a better average performance of CSGO algorithms (equipped with any chaotic map mentioned: Logistic, Double, Iterative, Singer and Cat) compared to the original SGO algorithm.

Considering the group of CSGO algorithms, the CSGO-Singer algorithm obtains the best solution (based on *Best Cost F* item), for all analyzed cases. Also, the CSGO-Logistic obtains better solutions than SGO (considering *Best cost F* item) for C1 and C5 cases. Additionally, CSGO algorithms perform better than other well-known algorithms (GA, PSO, DE, Jaya, TSA) or varieties of them, especially for the C1 system.

The obtained results, also confirmed by statistical tests, support the idea that including chaotic maps (Logistic, Iterative and Singer) in the structure of the original SGO algorithm can bring important benefits in solving ED problem.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The contributions of authors are as follows: Conceptualization, D.C. Secui, G. Bendea and M.L. Secui; methodology, D.C. Secui and M.L. Secui; software, data curation, D.C. Secui; validation, C. Hora, G. Bendea; formal analysis, C. Hora, D.C. Secui, investigation, G. Bendea, M.L. Secui; resources, D.C. Secui, M.L. Secui; writing—original draft preparation, D.C. Secui, G. Bendea, M.L. Secui; writing—review and editing, visualization, C. Hora, C. Bendea; supervision, funding acquisition, G. Bendea.

References

- [1] L. Papageorgiou and E. Fraga, "A Mixed Integer Quadratic Programming Formulation for the Economic Dispatch of Generators with Prohibited Operating Zones", *Electric Power System Research*, Vol. 77, No. 10, pp. 1292-1296, 2007.
- [2] J. Nanda, L. Hari, and M. L. Kothari, "Economic Emission Load Dispatch with Line Flow Constraints Using a Classical Technique", *IEE Proceedings Generation, Transmission and Distribution*, Vol. 141, No. 1, pp. 1-10, 1994.
- [3] A. I. Selvakumar and K. Thanushkodi, "A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems", *IEEE Transactions on Power Systems*, Vol. 22, No. 1, pp. 41-51, 2007.
- [4] A. K. Barisal, "Dynamic Search Space Squeezing Strategy Based Intelligent Algorithm Solutions to Economic Dispatch with Multiple Fuels", *International Journal of Electrical Power & Energy Systems*, Vol. 45, No. 1, pp. 50-59, 2013.
- [5] T. Jayabarathi, R. T. Kolipakula, M. V. Krishna, and A. Yazdani, "Application and Comparison of PSO, its Variants and HDE Techniques to Emission/Economic Dispatch", *Arabian Journal for Science and Engineering*, Vol. 39, No. 2, pp. 967-976, 2014.
- [6] Y. Labbi, D. Attous, and B. Mahdad, "Artificial Bee Colony Optimization for Economic Dispatch with Valve Point Effect", *Frontiers in Energy*, Vol. 8, No. 4, pp. 449-458, 2014.
- [7] H. M. Awais, T. N. Malik, and A. Ahmad, "Artificial Algae Algorithm with Multi-Light Source Movement for Economic Dispatch of Thermal Generation", *Mehran University Research Journal of Engineering and Technology*, Vol. 39, No. 3, pp. 564-582, 2020.
- [8] K. Bhattacharjee and N. Patel, "A Comparative Study of Economic Load Dispatch using Sine Cosine Algorithm", *Scientia Iranica*, Vol. 27, No. 3, pp. 1467-1480, 2020.
- [9] M. Suman and V. P. Sakhivel, "Coulomb's and Franklin's Laws Based Optimization for Nonconvex Economic and Emission Dispatch Problems", *International Energy Journal*, Vol. 20, No. 2, pp. 225-238, 2020.
- [10] M. Hassan, S. Kamel, L. Abualigah, and A. Eid, "Development and Application of Slime Mould Algorithm for Optimal Economic Emission Dispatch", *Expert Systems with Applications*, Vol. 182, pp. 1-28, 2021.
- [11] M. Suman, V. P. Sakhivel, and P. D. Sathya, "Squirrel Search Optimizer: Nature Inspired Metaheuristic Strategy for Solving Disparate Economic Dispatch Problems", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 111-121, 2020.
- [12] R. H. A. Rubayi, M. K. Abd, and F. M. F. Flaih, "A New Enhancement on PSO Algorithm for Combined Economic-Emission Load Dispatch Issues", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 1, pp. 77-85, 2020.

- [13] X. Chen, "Novel Dual-Population Adaptive Differential Evolution Algorithm for Large-Scale Multi-Fuel Economic Dispatch with Valve-Point Effects", *Energy*, Vol. 203, pp. 1-15, 2020.
- [14] M. A. Kaabi and L. A. Bahrani, "Modified Artificial Bee Colony Optimization Technique with Different Objective Function of Constraints Optimal Power Flow", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 378-388, 2020.
- [15] J. T. Yu, C. H. Kim, and S. B. Rhee, "Clustering Cuckoo Search Optimization for Economic Load Dispatch Problem", *Neural Computing and Applications*, Vol. 32, No. 22, pp. 16951-16969, 2020.
- [16] M. Pandit, L. Srivastava, M. Sharma, H. M. Dubey, and B. K. Panigrahi, "Large Scale Multi-zone Optimal Power Dispatch Using Hybrid Hierarchical Evolution Technique", *The Journal of Engineering*, Vol. 2014, No. 3, pp. 71-80, 2014.
- [17] S. Khamsawang and S. Jiriwibhakorn, "DSPSO-TSA for Economic Dispatch Problem with Nonsmooth and Noncontinuous Cost Functions", *Energy Conversion and Management*, Vol. 51, No. 2, pp. 365-375, 2010.
- [18] M. A. A. Betar, M. A. Awadallah, and M. M. Krishan, "A Non-Convex Economic Load Dispatch Problem with Valve Loading Effect Using a Hybrid Grey Wolf Optimizer", *Neural Computing and Applications*, Vol. 32, No. 16, pp. 12127-12154, 2020.
- [19] M. Kumar and J. S. Dhillon, "A Conglomerated Ion-Motion and Crisscross Search Optimizer for Electric Power Load Dispatch", *Applied Soft Computing*, Vol. 83, pp. 1-29, 2019.
- [20] B. A. Hassan, "CSCF: A Chaotic Sine Cosine Firefly Algorithm for Practical Application Problems", *Neural Computing and Applications*, Vol. 33, No. 12, pp. 7011-7030, 2021.
- [21] K. H. Truong, P. Nallagownden, Z. Baharudin, and D. N. Vo, "A Quasi-Oppositional-Chaotic Symbiotic Organisms Search Algorithm for Global Optimization Problems", *Applied Soft Computing Journal*, Vol. 77, pp. 567-583, 2019.
- [22] J. Yu, C. H. Kim, A. Wadood, T. Khurshid, and S. B. Rhee, "A Novel Multi-Population Based Chaotic Jaya Algorithm with Application in Solving Economic Load Dispatch Problems", *Energies*, Vol. 11, No. 8, pp. 1-25, 2018.
- [23] T. Guesmi, A. Farah, I. Marouani, B. Alshammari, and H. H. Abdallah, "A New Chaotic Sine Cosine Algorithm for Chance-Constrained Economic Emission Dispatch Problem Including Wind Energy", *IET Renewable Power Generation*, Vol. 14, No. 10, pp. 1808-1821, 2020.
- [24] D. C. Secui, "A Modified Symbiotic Organisms Search Algorithm for Large Scale Economic Dispatch Problem with Valve-Point Effects", *Energy*, Vol. 113, pp. 366-384, 2016.
- [25] S. Satapathy and A. Naik, "Social Group Optimization (SGO): A New Population Evolutionary Optimization Technique", *Complex & Intelligent Systems*, Vol. 2, No. 3, pp. 173-203, 2016.
- [26] A. Naik and S. Satapathy, "A Comparative Study of Social Group Optimization with a Few Recent Optimization Algorithms", *Complex & Intelligent Systems*, Vol. 7, No. 1, pp. 249-295, 2021.
- [27] C. L. Chiang, "Improved Genetic Algorithm for Power Economic Dispatch of Units with Valve-Point Effects and Multiple Fuels", *IEEE Transactions on Power Systems*, Vol. 20, No. 4, pp. 1690-1699, 2005.
- [28] A. H. Gandomi and X. S. Yang, "Chaotic Bat Algorithm", *Journal of Computational Science*, Vol. 5, No. 2, pp. 224-232, 2014.
- [29] C. Yang, S. W. Tsai, L. Chuang, and C. Yang, "An Improved Particle Swarm Optimization with Double-Bottom Chaotic Maps for Numerical Optimization", *Applied Mathematics and Computation*, Vol. 219, No. 1, pp. 260-279, 2012.
- [30] Cat Map, Wikipedia Web Site, http://en.wikipedia.org/wiki/Arnold%27s_cat_map (retrieved 15.08.2021).
- [31] J. T. Yu, C. H. Kim, A. Wadood, T. Khurshid, and S. B. Rhee, "Jaya Algorithm with Self-Adaptive Multi-Population and Lévy Flights for Solving Economic Load Dispatch Problems", *IEEE Access*, Vol. 7, pp. 21372-21384, 2019.
- [32] H. Lu, P. Sriyanyong, Y. H. Song, and T. Dillon, "Experimental Study of a New Hybrid PSO with Mutation for Economic Dispatch with Non-Smooth Cost Function", *International Journal of Electrical Power & Energy Systems*, Vol. 32, No. 9, pp. 921-935, 2010.
- [33] C. Chen, D. Zou, and C. Li, "Improved Jaya Algorithm for Economic Dispatch Considering Valve-Point Effect and Multi-Fuel Options", *IEEE Access*, Vol. 8, pp. 84981-84995, 2020.
- [34] K. Vaisakh and A. S. Reddy, "MSFLA/GHS/SFLA-GHS/SDE Algorithms for Economic Dispatch Problem Considering Multiple Fuels and Valve Point Loadings",

Applied Soft Computing Journal, Vol. 13, No. 11, pp. 4281-4291, 2013.

- [35] A. Meng, J. Li, and H. Yin, “An Efficient Crisscross Optimization Solution to Large-Scale Non-Convex Economic Load Dispatch with Multiple Fuel Types and Valve-Point Effects”, *Energy*, Vol. 113, pp. 1147-1161, 2016.
- [36] Z. X. Gang, L. Ji, M. Jin, and Z. Ying, “An Improved Quantum Particle Swarm Optimization Algorithm for Environmental Economic Dispatch”, *Expert Systems with Applications*, Vol. 152, pp. 1-14, 2020.