Hacettepe Journal of Mathematics & Statistics

Hacet. J. Math. Stat.
Volume 48 (5) (2019), 1570 – 1589
DOI : 10.15672/hujms.507579

Research Article

Implementation of improved grasshopper optimization algorithm to solve economic load dispatch problems

Muhammad Sulaiman*[1], Masihullah[1], Zubair Hussain[1], Sohail Ahmad[1], Wali Khan Mashwani[2], Muhammad Asif Jan[2], Rashida Adeeb Khanum[3]

[1] *Department of Mathematics, Abdul Wali Khan University Mardan, KP, Pakistan*
[2] *Department of Mathematics, Kohat University of Science and Technology, KP, Pakistan*
[3] *Jinnah College for Women, University of Peshawar, Pakistan*

Abstract

The costs of different fuels are increasing gradually, for operation of power production units. Thus new optimization techniques are needed to tackle the complex problems of Economic Load Dispatch (ELD). Metaheuristics are very helpful for policy and decision makers in achieving the best results by minimizing the cost function. In this paper, we have updated the Grasshopper Optimization Algorithm (GOA) with a better initialization strategy to balance the search capability of GOA. The new algorithm is named as Improved Grasshopper Algorithm (IGOA). GOA is inspired by the swarms of grasshopper and mimics their biological behavior. Furthermore, IGOA is used to solve the ELD problems by tacking four case studies from literature. The objective in these problems is to find best decision variables for dispatching the available power with lowest cost, better efficiency and more reliability. To validate the efficiency of our proposed algorithm, we have tested it by solving 4 case studies of ELD with 1263MW, 600MW, 800MW and 2500MW demands respectively. IGOA is better in terms of convergence rate and quality of solutions obtained for the problems considered in literature for other metaheuristics.

Keywords. constrained optimization, metaheuristics, improved grasshopper optimization algorithm (IGOA), economic load dispatch

1. Introduction

In the operation of power production plants, fuel costs play an important role and therefore scientists try to optimally dispatch the required load to their users. In problem of ELD minimum costs of power generations are determined to meet the required demand in a given time interval. The problem of ELD involves several constraints to determine the lowest cost of operation. Thus by minimizing the total cost of operation we achieve

*Corresponding Author.
Email addresses: sulaiman513@yahoo.co.uk (M. Sulaiman), masihullahbj@gmail.com (Masihullah), zubair.bsmaths@gmail.com (Z. Hussain), sohailahmadpk99@yahoo.com (S. Ahmad), mashwanigr8@gmail.com (W. K. Mashwani), majan.math@gmail.com (M. A. Jan), rakhan@uop.edu.pk (R. A. Khanum)
Received: 03.01.2019; Accepted: 13.05.2019

a balance in power production and dispatch to its users by significantly reducing the transmission losses.

In earlier studies, the problem of ELD was modeled as a single quadratic function and was solved using lambda iteration method, gradient based methods, as in [32]. Generally, these approaches have hitches in finding an overall optimum, usually offering local optimum solution only. Furthermore, traditional approaches require priori information regarding the continuity and differentiability of objective function belonging to the given optimization model. To overcome these shortcomings, quite a lot of nature inspired optimization techniques were designed and implemented. Particle swarm optimization [10] is one of the famous meta-heuristics applied to solve ELD problem. Other approaches used for solving ELD problems are evolutionary programming (EPs) [26], tabu search and multiple tabu search (TS, MTS)[20], differential evolution (DE)[16, 17], hybrid DE (DEPSO) [27], artificial bee colony algorithm (ABC) [12], simulated annealing (SA) [4], biogeography-based optimization [5], genetic algorithms [33], intelligent water drop algorithm [21], harmony search (HS) [9], hybrid harmony search [18], differential HS (DHS) [31], gravitational search algorithm [28], firefly algorithm [34], hybrid gravitational search [7] and cuckoo search (CS) [3] have been successfully applied to ELD problems.

As stated in theorem of no free lunch [11], that a single optimization technique cannot handle all types of optimization problems, which leads to the designing of new and updated optimization algorithms. Grasshopper optimization algorithm (GOA) [24] is published in 2017, which simulates the idea of the way grasshopper swarms search for their food. This paper aims at an improvement introduced into the GOA to solve ELD problems efficiently and prevent the algorithm from getting trapped in local optima. The Improved Grasshopper Optimization Algorithm (IGOA) is based on a novel initialization strategy. In order to test the efficiency of IGOA, we have compared our statistical results with its earlier version GOA and other state-of-the-art algorithms, including, GA, ESO, PSO, DE, HS, HHS.

The rest of this paper is organized as follows; section 2 demonstrates the ELD problem formulation considering objective function, generation limit constraint and power balance constraint. In section 3, the Grasshopper Optimisation Algorithm (GOA) and Improved GOA are described. In section 4, simulation results are presented that illustrate the potential of the proposed algorithm. Finally, section 6 concludes the paper.

2. Problem formulation

In the following section a detailed mathematical model for the problem of economic load dispatch is presented.

2.1. Objective function

The objective function in ELD is to minimize the costs incurred due to the generation of energy, which is a summation of all fuel costs of energy generated by all units. The mathematical form of the objective function is given in Equation 2.1 below:

$$Min\ C_t = \sum_{i=1}^{n} \alpha_i + \beta_i M_i + \gamma_i M_i^2, \tag{2.1}$$

where C_t denotes the total cost of energy production and n denotes power units involved in power generation. The constant coefficients in the objective function are denoted by the symbols α_i, β_i, γ_i and M_i represents the output of the i^{th} unit. By including the valve point effects, we get following objective function:

$$Min\ C_t = \sum_{i=1}^{n} \alpha_i + \beta_i M_i + \gamma_i M_i^2 + \left|\delta_i \sin(\epsilon_i(M_i^{min} - M_i))\right|, \tag{2.2}$$

where δ_i and ϵ_i are leading coefficients of the valve points for each unit. M_i^{min} is the lower generation limit of the i^{th} unit.
Practically, several types of fuels may be used for a generation unit. For a unit with N^{th} fuel option, Equation (2.1) is modified as:

$$C_{Ti} = \begin{cases} \alpha_{i1} + \beta_{i1} p_i + \gamma_{i1} M_i^2 & fuel \;\; 1 \;\; M_i^{min} \leq M_i \leq M_{i1}, \\ \alpha_{i2} + \beta_{i2} p_i + \gamma_{i2} M_i^2 & fuel \;\; 2 \;\; M_i^{min} \leq M_i \leq M_{i2}, \\ \vdots \\ \vdots \\ \alpha_{iN} + \beta_{iN} p_i + \gamma_{iN} M_i^2 & fuel \;\; N \;\; M_i^{min} \leq M_i \leq M_{iN}, \end{cases} \tag{2.3}$$

where C_{Ti} denotes the objective function for the cost of fuel of the i^{th} unit. The coefficients α_{iN}, β_{iN} and γ_{iN} are cost coefficients of the i^{th} generation unit operating on N type of fuel. Power units installed in multiple areas leads to the problem of Multi-area Economic Dispatch (MAED) Problem, where the objective function is a total of costs incurred by generating units in all areas. Thus,

$$Min \; C_T = \sum_{i=1}^{Na} C_{Ti}, \tag{2.4}$$

where C_{Ti} is the total cost of the i^{th} area and Na is the number of areas.

2.2. Constraints

Constraints imposed on the cost objective in ELD problems are given in the following sections:

2.2.1. Generation limit constraint. Each generating unit has a power production capacity which is given as,

$$m_i^{min} \leq m_i \leq m_i^{max}. \tag{2.5}$$

2.2.2. Power balance constraint.

$$\sum_{i=1}^{n} M_i = M_D + M_L, \tag{2.6}$$

where M_D and M_L denote the demand and transmission of network losses. For calculation of network losses, the B-coefficient method [6] is generally implemented by power production industries. In the B-coefficient method, the losses due to transmission lines is presented as a quadratic function as given below,

$$M_L = \sum_{i=1}^{n} \sum_{j=1}^{n} M_i B_{ij} M_j + \sum_{i=1}^{n} B_{0i} M_i + B_{00}. \tag{2.7}$$

3. Grasshopper optimisation algorithm (GOA)

Exploration and exploitation are the two major tendencies of those algorithms which happens to be initialized randomly. This method has been likened to the mood of life of grasshopper, which has several modes of life and shapes as shown in Figure 1. Being gregarious by nature, grasshopper has the social behavior of collective living. Not only in adult mature life, but even in the nymph stage several grasshopper live together. Popping up and down, they devour almost any vegetation, and in the aerial mode, they again swarm together in great coveys. Slow, steady movement and steps are the distinguishing features of the larvae when passing through metamorphosis. In total contrast to this mode of locomotion, when grasshopper attains to adult life the slow, steady locomotion is changed for jumping and random motion. In addition to these features and peculiarities, search for food source is another prominent feature of the insect [23]. During exploration the searchers ought to move in a random, and unsteady manner. But on the other hand, local thoughtful motion of grasshoppers fulfils the process of exploitation. A new, original algorithm, nature-inspired and intuitive called GOA has modelled this behavior mathematically. This new model employs this grouping behavior of grasshopper as shown in [29].

$$X_i = S_i + G_i + A_i, \tag{3.1}$$

where X_i is i^{th} grasshopper's position during locomotion, S_i denotes social interaction between the grasshoppers, G_i is the force of gravity on the i^{th} grasshopper, A_i shows vertical motion in the air.

For random behaviour, the above equation can be written as $X_i = r_1 S_i + r_2 G_i + r_3 A_i$, where r_1, r_2, and r_3 are random numbers in $[0, 1]$. We get a formula for social interaction as,

$$S_i = \sum_{j=1,\ j\neq i}^{n} s(d_{ij})\widehat{d_{ij}}, \tag{3.2}$$

where d_{ij} is the distance measured between the i^{th} and the j^{th} grasshopper, while $d_{ij} = |x_j - x_i|$, s is a function which denote the strength of social forces between these grasshoppers, as shown in Equation (3.3), $\widehat{d_{ij}} = \frac{x_j - x_i}{d_{ij}}$ it depicts unit vector starting from the i^{th} grasshopper to the j^{th} grasshopper. The function s, defining social forces can be calculated as :

$$s(r) = fe^{\frac{-r}{l}} - e^{-r}, \tag{3.3}$$

where f show the intensity of the mutual attraction between the grasshopper and l is the attractive length scale.

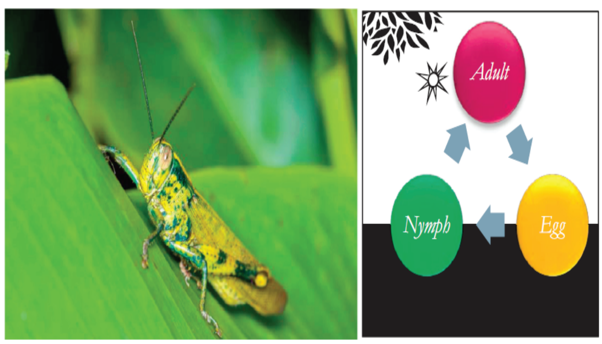

Figure 1. Real grasshopper and life cycle of grasshoppers, [24].

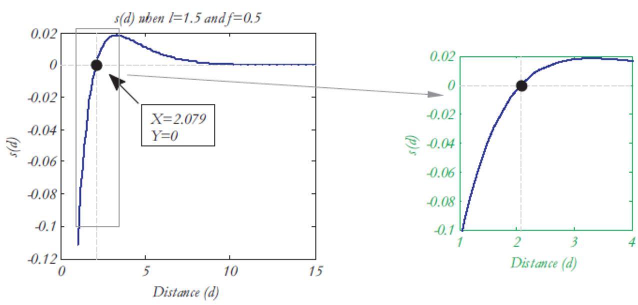

Figure 2. left) Function s when $l = 1.5$ and $f = 0.5$ (right) Range of function s when x is in $[1, 4]$, [24].

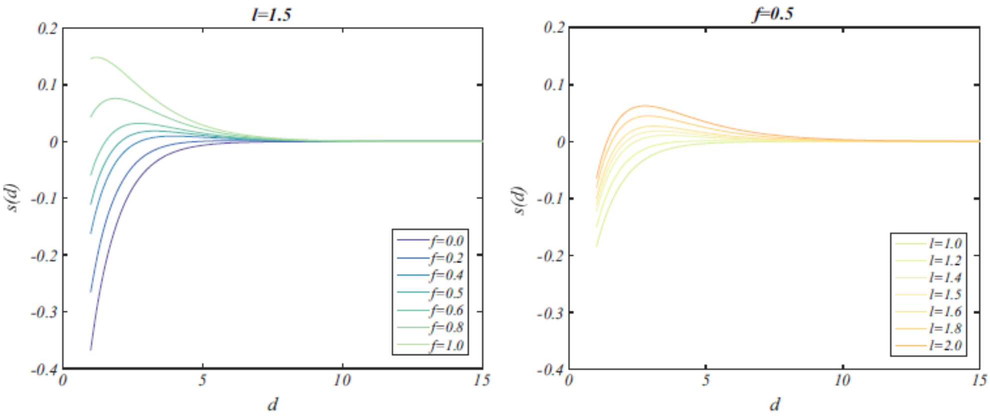

Figure 3. Behaviour of the function s when varying l or f, [14]

To show its impact on the mutual interaction of grasshoppers the following Figure 2 is very helpful. The graph shows distance from 0 to 15, and the units are -0.12 to 0.02. At point (0, 2.079) repulsion occurs. The graph shows that when a particular grasshopper is at a distance of 2.079 units from anther grasshopper, the attraction and repulsion among the grasshopper are mutually balanced, and known as comfort zone. As evident from the graph, attraction is on the increase from 2.079 unit up to nearly 4, and then decline starts. By changing the l and f parameters in Equation (3.3), it denotes different social behaviours of grasshoppers. By varying l and f independently, a graph of the function s has been drawn in Figure 3. This clearly depicts the effects of these parameters, l and f change significantly the comfort zone, repulsion region as well as the attraction region. For very small values of l and f attraction and repulsion regions happens to be very small. From all these values we have chosen $l = 1.5$ and $f = 0.5$. The function s, as illustrated in Figure 4, depicts the conceptual model of the mutual interaction of grasshoppers and comfort zone. As Figures 2 and 3 depict, function s has the ability to divide the space between the two grasshoppers into three regions i.e. repulsion region, attraction region and comfort zone [13]. When the distance is greater than 10 between the two grasshoppers, the result become zero. It is clear that this function S has no ability for application of strong forces, when the grasshoppers are distant from one another. In order to resolve this problem the distance of grasshopper in [1,4] interval has been graphed as shown in the Figure 2, at the right side. We can calculate the G component as in Equation (3.1),

$$G_i = -g\widehat{e_g}, \tag{3.4}$$

in this equation g denotes gravitational constant and $\widehat{e_g}$ depicts a unity vector, pointing in the direction of the centre of earth. Similarly, we can calculate the component A in Equation (3.1) as,

$$A_i = u\widehat{e_w}, \tag{3.5}$$

in this equation, u denotes a constant drift, and it is a unity vector pointing in the direction of velocity of wind. Now by substituting the respective value of S, G, and A in Equation (3.1)

$$X_i = \sum_{j=1,\ j\neq i}^{n} s(X_j - X_i)\frac{X_j - X_i}{d_ij} - g\widehat{e_g} + u\widehat{e_w}, \tag{3.6}$$

this is the expanded form of the Equation (3.1). No sooner the nymphs starts to trod on the land, there ought to be a limit to their position. Equation (3.6) can be used as simulation for the interaction between grasshoppers in swarm only. By using this Equation (3.6) behaviour of two different insect swarms in 2-dimension and 3-dimensions have been graphically drawn in Figures 7 and 8. It is prerequisite for this kind of experiment that

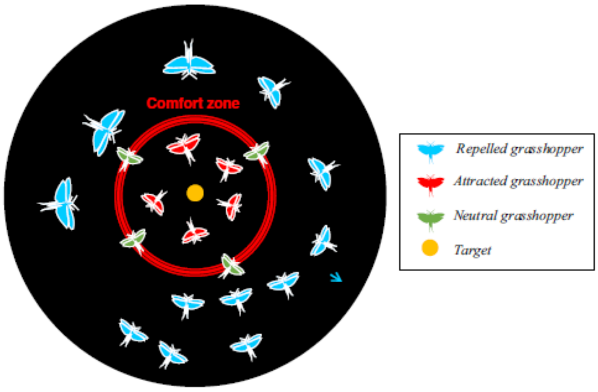

Figure 4. Attraction, repulsion and comfort zone among grasshoppers, [1].

almost 20 artificially created grasshopper move overtime duration of 10 units.

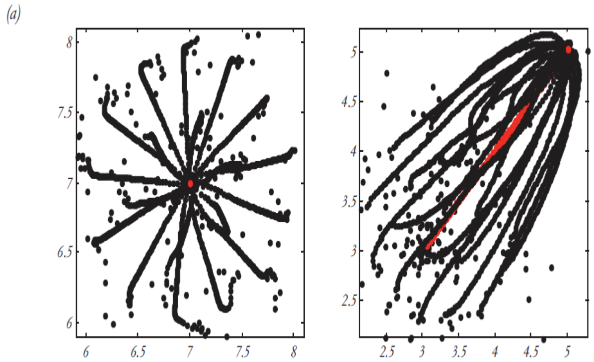

Figure 5. Behaviour of swarm in a 2D space, [24].

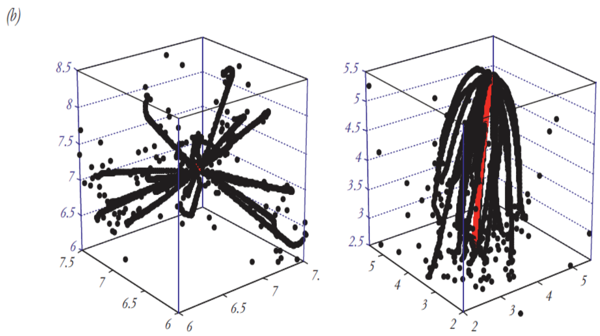

Figure 6. Behaviour of swarm in a 3D space [24].

Equation (3.6) tries to bring closer together the initially disordered population into well-knit and well ordered swarm. This is shown in the 2-dimensional Figure 5. After lapse of 10 units of time, the comfort zone is attained by the grasshopper and then cease to move anymore. Likewise Figure 6, is a depiction of this same behavior in 3-dimensions. It is evident from these facts that the mathematical model is fully capable of simulating a covey of grasshoppers in 2, 3 or higher dimensions. In spite of so much close simulation,

optimization problems cannot be directly solved by this mathematical model. The main reason is that the grasshopper attain the comfort zone in no time and the locust group's concentration in to a specified point is not possible.
Considering this short coming, a modified version of Equation (3.6) has been proposed which is able to solve optimization problems by producing better solution. This proposed equation is the following;

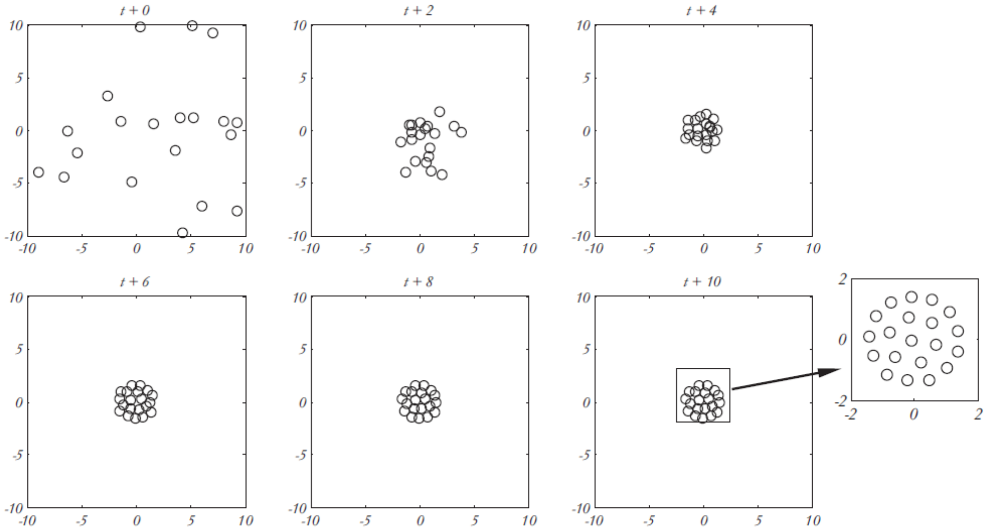

Figure 7. Behaviour of swarm in a 2D space, [24].

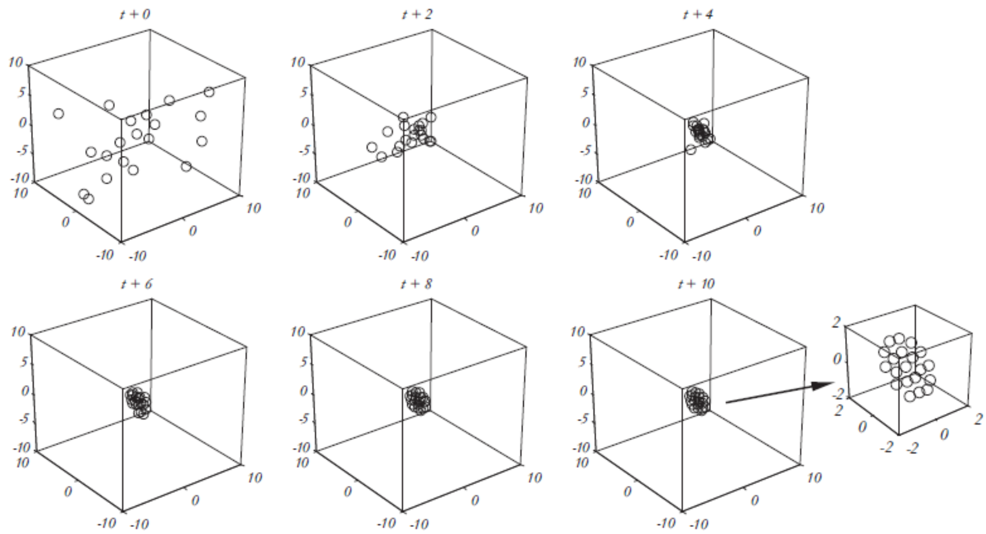

Figure 8. Behaviour of swarm in a 3D space, [24].

$$X_i^d = c\left(\sum_{j=1,\ j\neq i}^{n} c\frac{ub_d - lb_d}{2} s\left(\left|X_j^d - X_i^d\right|,\right)\frac{X_j - X_i}{d_{ij}}\right) + \widehat{T_d}, \tag{3.7}$$

where ub_d is the upper bound in the D^{th} dimension, lb_d is the lower bound in the D^{th} dimension $s(r) = fe^{\frac{-r}{l}} - e^{-r}$, $\widehat{T_d}$ is the value of the D^{th} dimension in the target (best solution found so far), and c is a decreasing coefficient to shrink the comfort zone, repulsion zone, and attraction zone. Note that S is almost similar to the S component in Equation (3.1). In this equation gravity (G) component has been removed. Moreover, it has been assumed that the wind flows in the direction of the target i.e. $\widehat{T_d}$.

Looking at the Equation (3.7), the grasshopper's next position depend on its present position, as well as the position of the other grasshoppers. In order to know for certain the search agents location all around the target, status of all the grasshoppers have been considered.

In contrast with PSO where two vectors for each particle, i.e. position and velocity vector are used for its location and specification, in GOA only one vector i.e. of position vector is used. Moreover, GOA has a slight edge over PSO in connection with the updating the position of a search agent.

In PSO only current position, global best and personal best are taken into consideration, while in case of GOA. Besides these, position of all other search agents are also taken into consideration. So it is clear from this that in case of PSO, other particles are not involved for updating the particle's position, but on the other hand in the case of GOA, all other search agents are involved to define the location of every one of the search agent. The main reasons behind inserting the adaptive parameter c in Equation (3.7) are these. Inertial weight w in PSO and c in Equation (3.7) are similar things. It is responsible for slowing down the grasshopper's movements around the target, i.e. balance the exploration and exploitation of all the grasshopper (any other searcher) all around the targeted region. The other c, incorporated in the equation, is responsible for decreasing the attraction, comfort and repulsion zones between the grasshoppers. The component $c\frac{ub_d-lb_d}{2}s(|X_j - X_i|)$ in Equation (3.7) has greatly enhanced the work ability of the model. The space which the grasshopper explore and exploits greatly diminished in a linear way by the incorporation of the term, $c\frac{ub_d-lb_d}{2}$. Moreover, the component $s(|x_j - x_i|)$ has been inserted in order to determine whether to repel the grasshopper from or attract towards the target during exploration and exploitation stages respectively.

The component c which occupies almost the middle of the equation serves to mitigate considerably the repulsive and attractive forces between different grasshopper individuals. It is taken as proportional to number of times the repulsion and attraction occurs. On the other hand, the term c which occupies the outer position, serves to reduce the search coverage when the iteration counting increases much more. Briefly speaking, the component in the above Equation (3.7) takes into consideration the location of the other grasshoppers and active interaction of the grasshopper with nature. The function of the second term, $\widehat{T_d}$ is the moving tendency of the grasshopper towards the food source. Likewise, parameter c also simulates reduction in the velocity of the grasshopper, while approaching food source and devouring it. In order to get random behaviour in both interaction with other grasshopper or nature, as well as to show special inclination towards food source, individual terms should be multiplied with random valves. This new formulation has enabled the researcher to fully explore and then exploit the specific research space. For tuning the level of exploration to the level of exploitation the search agents must have some mechanism at their disposal. In the larval stage of their life cycle, grasshoppers at first move at in search of food near about their habitat, but once the wings are competed, they make air flights of varying scales, land explore new sources of food.

Stochastic optimization algorithms has the scheme that exploration is started prior exploitation. It is due to the need for to search out some search space, holding chances of food source. Search agents are obliged to search locally to find out global optimum to a very near approximation. In order to make a balance between exploration and exploitation the c parameter is decreased in proportion to the number of repetition.
When with passage of time the iteration count increases, this kind of mechanism greatly promote the exportation. In proportion to the number of iterations, the coefficient c

greatly reduces the comfort zone. It is calculated in the following manner:

$$c = cmax - l\frac{cmax - cmin}{L}, \tag{3.8}$$

where *cmax* is the maximum value, *cmin* is the minimum value, l indicates the current iteration, and L is the maximum number of iterations.
In this paper, we use 1 and 0.00001 for cmax and cmin respectively. The parameter c, greatly affects the locomotion and tendency of convergence of the grasshopper as shown in Figures 5 and 6. More than 90 iterations have been taken and the sub-figures accompanying these figures, well illustrate the positions of the grasshoppers.

Although in this model grasshoppers are made to move towards a target in a gradual manner over the span of several iterations, but in the realm of real search space, no target exists as we cannot know where the global best solution is located. So in each individual step of the optimization, new target for the grasshopper must be sought out. This piecewise optimization is of great assistance for the GOA. Through it the most suitable target in the search space is solved and the real global optimization is attained.

3.1. Improved GOA

The quality of population is an important factor which can directly or indirectly effect the strength of an algorithm in searching the given domain for an optimal solution. Also having an initialization process with random generation of candidate solutions is not an effective idea in every case, specially when the search space is large. Hence we have updated the GOA by dividing the capabilities of the algorithm in two parts, as shown in Figure 9, which is named as Improved GOA or IGOA.

In first part, the algorithm initializes with a fixed random population for certain number of evaluations, using Equation 3.9,

$$U_i = Lb_i + (Ub_i - Lb_i) * rand(0, 1), \tag{3.9}$$

In the second part, the algorithm is focused and initialized with the best so far spots found during the earlier evaluations. This strategy is shown to be very efficient in getting the best results with the less number of function evaluations and time to solve the problems.

4. Results and discussion

In the following section, we have tested IGOA by solving 4 standard case studies of ELD as in [15]. It is obvious from literature, that many researchers have solved these case studies by using other optimization techniques as in [30]. We have implemented IGOA to solve these problems and our results are compared with other standard algorithms.

4.1. Experimental settings

To check the robustness of our technique we have repeated our simulations 100 times, size of population was fixed as 40, and total number of generations were limited to 100. Our results are compared with GA, ESO, DE, HS, HHS, FFA, BBO, LI, HM, ALO. Moreover, details about the four case studies are given in the following sections:

4.2. Case study 1

This system contains 6 thermal generators and power demand is 1263 MW. The prohibited working zones are given in Table 1. Best objective values are presented in Table 3. In Table 4 performances of different algorithms are compared, such as, Grasshopper Optimization Algorithm (GOA), Improved Grasshopper Optimization Algorithm (IGOA),

```
if N.E =< limit     % number of Evaluation
Initialize the swarm X_i ( i = 1, 2, 3, ... ... , n ), randomly.
else
Select n-best so far solutions obtained in earlier evaluations. And built a new population.
endif
    Calculate the fitness each search agent
    T= best search agent
    while (l < Max number of iterations)
        Update c using Equation (3.8)
        for  Each search agent
            Normalize the distance between grasshoppers in [1,4]
            Update the position of the current search agent by the Equation (3.7)
            Bring the current search agent back if it goes outside the boundaries
        endfor
        Update T if there is a better solution
        l = l + 1
endwhile
Return T
```

Figure 9. Pseudocode of the IGOA algorithm.

Genetic Algorithms (GA) [35], Evolutionary Strategy Optimisation (ESO) [19], Differential Evolution (DE) [16], Particle Swarm Optimization (PSO) [10], Harmony Search (HS) [8] and Hybrid Harmony Search (HHS) [8]. It is observed that both GOA and IGOA performs better than the other methods in terms of the solution quality as shown in Figure 10. The characteristics of the generation units and the B-coefficients (with base capacity 100 MVA) for network losses are given in Table 2 and B_{ij}. The convergence characteristics of the proposed IGOA are plotted in Figure 11.

Table 1. Prohibited operating zones

Power unit	Prohibited zones	
01	[210 240]	[350 380]
02	[90 110]	[140 160]
03	[150 170]	[210 240]
04	[80 90]	[110 120]
05	[90 110]	[140 150]
06	[75 85]	[100 105]

Table 2. Data for 6 unit system

Unit	α_i	β_i	γ_i	M_{max}	M_{min}
01	240	7.0	0.0070	500	100
02	200	10.0	0.0095	200	50
03	220	8.5	0.0090	300	80
04	200	11.0	0.0090	150	50
05	220	10.5	0.0080	200	50
06	190	12	0.0075	120	50

$$B_{ij} = \begin{pmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & 0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0.0000 & -0.001 & -0.0006 \\ -0.0001 & 0.0001 & 0.0000 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.001 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.0150 \end{pmatrix},$$

$$B_{ij} = \begin{pmatrix} 0.001 \times (-0.3908 & -0.1297 & 0.7047 & 0.0591 & 0.2161 & -0.6635) \end{pmatrix},$$

$$B_{00} = 0.0560.$$

Table 3. Results of Case Study 1 for 1263MW total demand with power losses

Unit	GA	ESO	PSO	DE	HS	HHS	GOA	IGOA
M_1	474.81	447.50	451.56	447.74	449.381	447.496	493.8084	447.82
M_2	178.64	173.32	173.44	173.41	173.530	173.314	170.145	184.4384
M_3	262.21	263.48	263.99	263.41	263.524	263.445	252.5353	256.9527
M_4	134.28	139.06	147.46	139.08	132.049	139.055	117.581	114.0006
M_5	151.90	165.48	429.64	165.36	167.262	165.475	172.1718	179.8744
M_6	74.18	87.13	71.32	86.94	90.262	87.125	64.6851	88.52058
Total generation (MW)	1276.03	1275.96	1272.46	1275.95	1276.01	1275.91	1271	1271.607
$Power_{loss}$	13.02	12.96	12.82	12.96	13.08	12.95	8	8.607

Table 4. In the 100 trial tests, best results obtained by various algorithms (Case Study 1)

method	Generation cost($)			CPU time (s)
-	Max.	Min	Average	-
GA	15524.0	15459.0	15469.0	41.580[a]
PSO	15492.0	15450.0	15454.0	14.860[a]
ESO	15470.0	15408.0	15430.0	0.360[a]
DE	15450.0	15450.0	15450.0	0.0330[a]
HS	15449.0	15449.0	14449.0	6.830
HHS	15453.0	15449.0	15450.0	0.140
GOA	15412	**15401**	15406.5	0.50
IGOA	15408	**15393.92**	15401.42	0.42

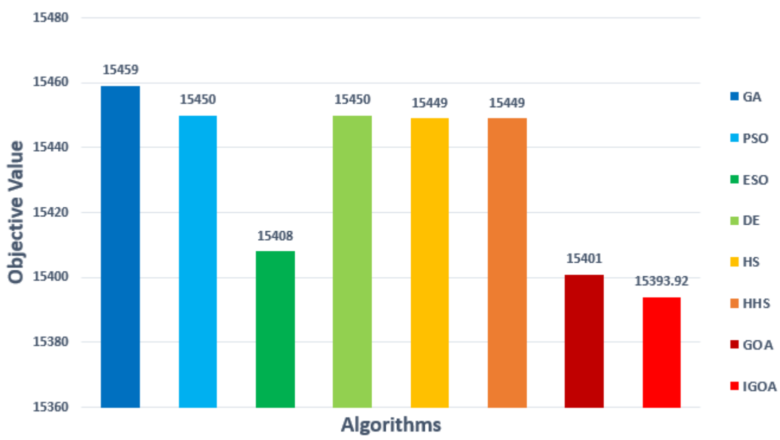

Figure 10. Best results obtained, for case study 1, by different techniques are compared with GOA and IGOA algorithms

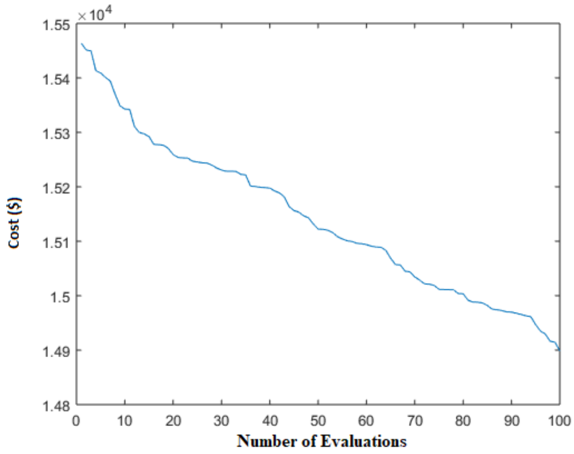

Figure 11. IGOA takes less than 100 function evaluations by getting better results for case study 1.

4.3. Case study 2 (3 generating units with load demand of 600MW)

This system contains 3 thermal generators and power demand is 600 MW [22]. The unit data and B-coefficients for network losses are given in Table 5. In Table 6 performances of different algorithms are compared, such as, Grasshopper Optimization Algorithm (GOA), Improved Grasshopper Optimization Algorithm (IGOA), Lambda-iteration method (LI) [22], Firefly Algorithm (FFA) [22] and Ant Lion Optimizer (ALO) [15]. The comparison between the generation cost is presented in Table 8. From our results it is obvious that GOA and IGOA obtained better or similar solutions as compared with other methods shown in Figure 12. The convergence characteristics of the proposed IGOA are plotted in Figure 13.

Table 5. Data for 3 unit system

Unit	α_i	β_i	γ_i	M_{min}	M_{max}
01	1243.5311	38.30553	0.03546	35	210
02	1658.5696	36.32782	0.02111	130	325
03	1356.6592	38.27041	0.01799	125	315

$$B_{ij} = \begin{pmatrix} 0.000071 & 0.000030 & 0.000025 \\ 0.000030 & 0.000069 & 0.000032 \\ 0.000025 & 0.000032 & 0.000080 \end{pmatrix},$$

Table 6. Optimal load dispatch for 3 unit system with power losses

Unit power output (MW)	Methods			
-	FFA	ALO	GOA	IGAO
M_1	130.021	130.02	165.92	155.893
M_2	250.84	250.85	245.685	251.11
M_3	236.43	236.44	202.383	207.0869
Total generation (MW)	617.291	617.31	614	614.0899
$Power_{loss}$	17.30	17.3040	14	14.0899

Table 7. In 100 trial tests, best results obtained by various algorithms (Case Study 2)

Method	Generation cost ($)
Lambda iteration	30359.3
FFA	30334.0
ALO	30333.9858
GOA	**30273**
IGOA	**30243**

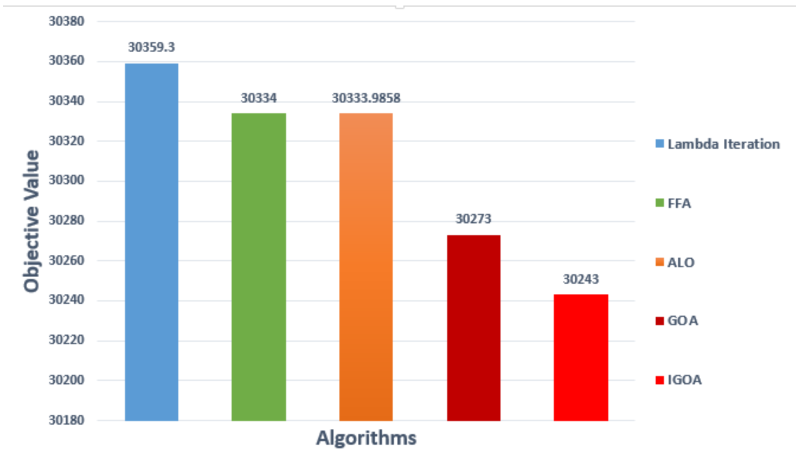

Figure 12. Best results obtained, for case study 2, by different techniques are compared with GOA and IGOA algorithms

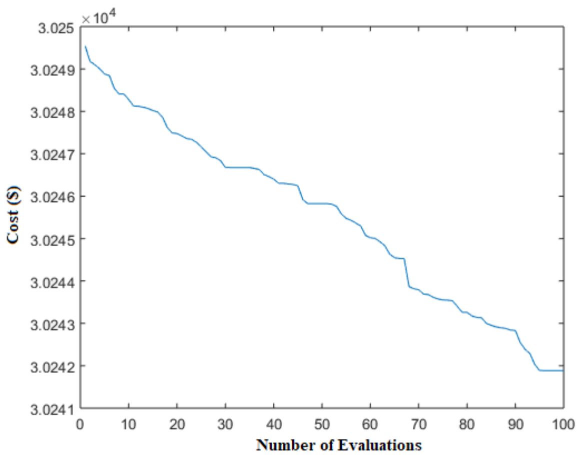

Figure 13. IGOA takes less than 100 function evaluations by getting better results for case study 2.

4.4. Case study 3 (6 generating units with power demand 800MW)

This system contains 6 thermal generators and power demand is 800MW [22]. Generation limits and B-coefficients matrix of this system are given in Table 8. In table 9 performances of different algorithms are compared, such as, Grasshopper Optimization

Algorithm (GOA), Improved Grasshopper Optimization Algorithm (IGOA), swarm optimization (PSO) [10], Lambda iteration method (LI) [22], Firefly Algorithm (FFA) [22] and Ant Lion Optimizer (ALO) [15]. The comparison between the generation cost is presented in Table 10. The outcome suggests that both GOA and IGOA are more suitable algorithm as compare to other algorithms in the given case study. Our results are better in terms of minimum cost as compered with other algorithms, shown in Figure 14. As shown in Figure 15, IGOA takes less number of function evaluations to get the solutions of required quality.

Table 8. Data for 6 unit system

Unit	α_i	β_i	γ_i	M_{min}	M_{max}
01	756.79886	38.53	0.15240	10	125
02	451.32513	46.15916	0.10587	10	150
03	1049.9977	40.39655	0.02803	35	225
04	1243.5311	38.30553	0.03546	35	210
05	1658.5696	36.32782	0.02111	130	325
06	1356.6592	38.27041	0.01799	125	315

The loss co-efficient matrix of 6-Unit system

$$B_{ij} = \begin{pmatrix} 0.0000220 & 0.00020 & 0.0000190 & 0.000025 & 0.0000320 & 0.000085 \\ 0.000026 & 0.0000150 & 0.000024 & 0.000030 & 0.000069 & 0.000032 \\ 0.000019 & 0.000016 & 0.000017 & 0.000071 & 0.000030 & 0.0000250 \\ 0.000015 & 0.000013 & 0.000065 & 0.000017 & 0.000024 & 0.000019 \\ 0.000017 & 0.000060 & 0.000013 & 0.000016 & 0.000015 & 0.000020 \\ 0.00014 & 0.000017 & 0.000015 & 0.000019 & 0.000026 & 0.000022 \end{pmatrix},$$

Table 9. Results of Case Study 3 for 800MW total demand with power losses

Unit power output (MW)	Methods				
	PSO	FFA	ALO	GOA	IGOA
M_1	32.599	32.5863	32.6003	38.769	47.60463
M_2	14.483	14.4843	14.4830	19.9360	35.36481
M_3	141.544	141.548	141.5440	55.825	94.712856
M_4	136.041	136.045	136.0413	132.3364	74.793288
M_5	257.658	257.664	257.6587	311.0588	306.3950
M_6	242.003	243.009	243.0033	262.54999	259.5692
$Power_{Loss}$	25.3306	25.3312	25.3307	20.476	18.43
Total	41896.62	41896.9	41896.6286	41868	41865

Table 10. In 100 trial tests, best results obtained by various algorithms (Case Study 3)

Method	Generation cost ($)
PSO	41896.66
Lambda iteration	41959.0
FFA	41896.9
ALO	41896.6286
GOA	**41868.00**
IGOA	**41865**

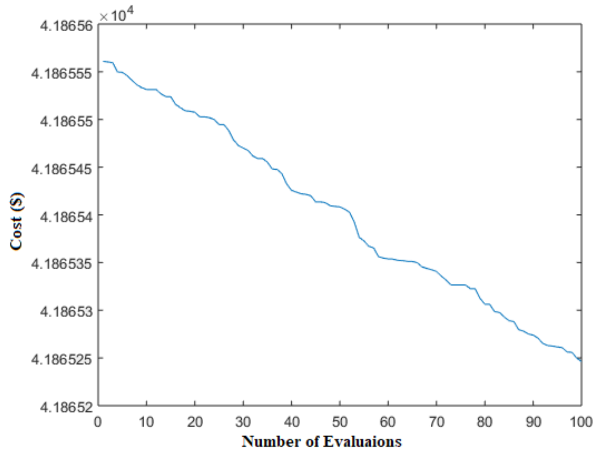

Figure 15. IGOA takes less than 100 function evaluations by getting better results for case study 3.

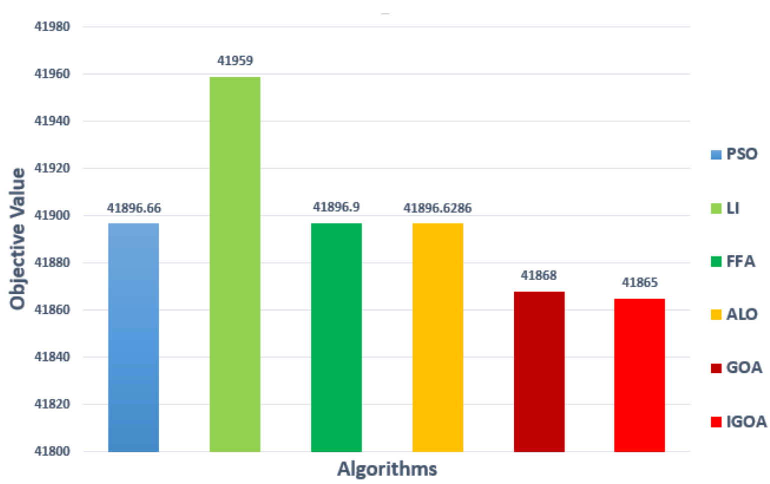

Figure 14. Best results obtained, for case study 3, by different techniques are compared with GOA and IGOA algorithm

4.5. Case study 4 (20 generating units with load demand of 2500MW)

This case study consists of 20 thermal generators and power demand is 2500 MW. The input data and B_{ij} i.e; the loss coefficient are taken from [25], and is given in Table 11. The obtained output is compared with Biogeography-Based Optimization (BBO) algorithm [2], Lambda iteration method (LI) [25], Hopfield modeling (HM) [25] and Ant Lion Optimizer (ALO) [15] in Table 12. The obtained results are better from other algorithms which are shown in comparison Table 13, Figure 16. As shown in Figure 17, IGOA takes less number of function evaluations to get the solutions of required quality.

Table 11. Data for 20 unit system

Unit	α_i	β_i	γ_i	M_{min}	M_{max}
1	0.00068	18.19	1000	150	600
2	0.00071	19.26	970	50	200
3	0.0065	19.80	600	50	200
4	0.00500	19.10	700	50	200
5	0.00738	18.10	420	50	160
6	0.00612	19.26	360	20	100
7	0.00790	17.14	490	25	125
8	0.00813	18.92	660	50	150
9	0.00522	18.27	765	50	200
10	0.00573	18.92	770	30	150
11	0.00480	16.69	800	100	300
12	0.00310	16.76	970	150	500
13	0.00850	17.36	900	40	160
14	0.00511	18.70	700	20	130
15	0.00398	18.70	450	25	185
16	0.07120	14.26	370	20	80
17	0.00890	19.14	480	30	85
18	0.00713	18.92	680	30	120
19	0.00622	18.47	700	40	120
20	0.00773	19.79	850	30	100

$$B_{ij} = \begin{pmatrix}
8.70 & 0.43 & -4.61 & 0.36 & 0.32 & -0.66 & 0.96 & -0.60 & 0.80 & -0.10 & 3.60 & 0.64 & 0.79 & 2.10 & 1.70 & 0.80 & -3.20 & 0.70 & 0.48 & -0.70 \\
0.43 & 8.30 & -0.97 & 0.22 & 0.75 & -0.28 & 5.04 & 1.70 & 0.54 & 7.20 & -0.28 & 0.98 & -0.46 & 1.30 & 0.80 & 0.20 & 0.52 & -.70 & 0.80 & 0.20 \\
-4.61 & -0.97 & 9.00 & -2.00 & 0.63 & 3.00 & 1.70 & -4.30 & 3.10 & -2.00 & 0.70 & -0.77 & 0.93 & 4.60 & -0.30 & 4.20 & 0.38 & 0.70 & -0.20 & 3.60 \\
0.36 & 0.22 & -2.0 & 5.30 & 0.47 & 2.62 & -1.96 & 2.10 & 0.67 & 1.80 & -0.45 & 0.92 & 2.40 & 7.60 & -0.20 & 0.70 & -1.00 & 0.86 & 1.60 & 0.87 \\
0.32 & 0.75 & 0.63 & 0.47 & 8.60 & -0.80 & 0.37 & 0.72 & -0.90 & 0.69 & 1.80 & 4.30 & -2.80 & -0.70 & 2.30 & 6.60 & 0.80 & 0.20 & -3.00 & 0.50 \\
-0.66 & -0.28 & 3.00 & 2.62 & -0.80 & 11.8 & -4.90 & 0.30 & 3.0 & -3.0 & 0.40 & 0.78 & 6.40 & 2.60 & -0.20 & 2.10 & -0.40 & 2.30 & 1.60 & -2.10 \\
0.96 & 5.04 & 1.70 & 1.96 & 0.37 & -4.90 & 8.24 & -0.90 & 5.90 & -0.60 & 8.50 & -0.83 & 7.20 & 4.80 & -0.90 & -0.10 & 1.30 & 0.76 & 1.90 & 1.30 \\
-0.60 & 1.70 & -4.30 & 2.10 & 0.72 & 0.30 & -0.90 & 1.20 & -0.96 & 0.56 & 1.60 & 0.80 & -0.40 & 0.23 & 0.75 & -0.56 & 0.80 & -0.30 & 5.30 & 0.80 \\
0.80 & 0.54 & 3.10 & 0.67 & -0.90 & 3.00 & 5.90 & -0.96 & 0.93 & -0.30 & 6.50 & 2.30 & 2.60 & 0.58 & -0.10 & 0.23 & -0.30 & 1.50 & 0.74 & 0.70 \\
-0.10 & 7.20 & -2.0 & 1.80 & 0.69 & -3 & -0.6 & 0.56 & -0.3 & 0.99 & -6.6 & 3.9 & 2.3 & -0.3 & 2.8 & -0.8 & 0.38 & 1.9 & 0.47 & -0.26 \\
3.6 & -0.28 & 0.7 & -0.45 & 1.8 & 0.4 & 8.5 & 1.6 & 6.5 & -6.6 & 10.7 & 5.3 & -0.6 & 0.7 & 1.9 & -2.6 & 0.93 & -0.6 & 3.8 & -1.5 \\
0.64 & 0.98 & -0.77 & 0.92 & 4.3 & 0.78 & -0.83 & 0.80 & 2.3 & 3.9 & 5.3 & 8.0 & 0.9 & 2.1 & -0.7 & 5.7 & 5.4 & 1.5 & 0.7 & o.1 \\
0.79 & -0.46 & 0.93 & 2.4 & -2.8 & 6.4 & 7.2 & -0.4 & 2.6 & 2.3 & -0.6 & 0.9 & 11 & 0.87 & -1 & 3.60 & 0.46 & -0.9 & 0.6 & 1.5 \\
2.1 & 1.3 & 4.6 & 7.6 & -0.7 & 2.6 & 4.8 & 0.23 & 0.58 & -0.3 & 0.7 & 2.1 & 0.87 & 3.8 & 0.5 & -0.7 & 1.9 & 2.3 & -0.97 & 0.9 \\
1.7 & 0.8 & -0.3 & -0.2 & 2.3 & -0.2 & -0.9 & 0.75 & -0.1 & 2.8 & 1.9 & -0.7 & -1.0 & 0.5 & 11.0 & 1.9 & -0.8 & 2.6 & 2.3 & -0.1 \\
0.8 & -0.2 & 4.2 & 0.7 & 3.6 & 2.1 & -0.1 & -0.56 & 0.23 & -0.8 & -2.6 & 5.7 & 3.6 & -0.7 & 1.9 & 10.8 & 2.5 & -1.8 & 0.9 & -2.6 \\
-3.2 & 0.52 & 0.38 & -1 & 0.8 & -0.4 & 1.3 & 0.8 & -0.3 & 0.38 & 0.93 & 5.4 & 0.46 & 1.9 & 0.8 & 2.5 & 8.7 & 4.2 & -0.3 & 0.68 \\
0.7 & -1.7 & 0.7 & 0.86 & 0.2 & 2.3 & 0.76 & -0.3 & 1.5 & 1.9 & -0.6 & 1.5 & -0.9 & 2.3 & 2.6 & -1.8 & 4.2 & 2.2 & 0.16 & -0.3 \\
0.48 & 0.8 & -2.0 & 1.6 & -3 & 1.6 & 1.9 & 5.3 & 0.74 & 0.47 & 3.8 & 0.7 & 0.6 & -0.97 & 2.3 & 0.9 & -0.3 & 0.16 & 7.6 & 0.69 \\
-0.7 & 0.2 & 3.6 & 0.87 & 0.5 & -2.1 & 1.3 & 0.8 & 0.7 & -0.26 & -1.5 & 0.1 & 1.5 & 0.9 & -0.1 & -2.6 & 0.68 & -0.3 & 0.69 & 7.0
\end{pmatrix},$$

Table 12. Optimal dispatch for Case Study 4 with power losses

Unit	BBO	LI	HM	ALO	GOA	IGOA
M1	513.08920	512.78050	512.78040	512.780	399.04232	600
M2	173.35330	169.10330	169.10350	169.110	149.73828	50.25150
M3	126.92310	126.88980	126.88970	126.890	50	50
M4	103.32920	102.86570	102.86560	102.870	82.281997	50
M5	113.77410	113.63860	113.68360	113.680	55.831	160
M6	73.066940	73.57100	73.57090	73.5680	79.635	20
M7	114.98430	115.28780	115.28760	115.290	120.601	125
M8	116.42380	116.39940	116.39940	116.40	65.84208	50
M9	100.69480	100.40620	100.440630	100.410	86.7785	50
M10	99.999790	106.02670	106.02670	106.020	48.3471	30
M11	148.97700	150.23940	150.23950	150.240	213.9353	300
M12	294.02070	292.76480	292.6770	292.770	500	500
M13	119.57540	119.11540	119.11550	119.120	103.4156	160
M14	30.547860	30.83400	30.83420	30.8310	121.2949	20
M15	116.45460	115.80570	115.80560	115.810	185	185
M16	36.227870	36.25450	36.25450	36.2540	50.1305	20
M17	66.859430	66.85900	66.85900	66.8570	77.456	30
M18	88.547010	87.97200	87.9720	87.9750	56.4623	120
M19	100.98020	100.80330	100.80330	100.80	76.7435	40
M20	54.27250	54.30500	54.3050	54.3050	69.34415	30
Total generation (MW)	2592.1011	2591.9670	2591.9670	2591.967	2591.88	2590.3
$power_{loss}$	92.1011	91.9670	91.9669	91.9662	91.88	90.3

Table 13. In 100 trial tests, best results obtained by various algorithms (Case Study 4)

Method	Generation cost ($)
BBO	62456.77926
LI	62456.6391
HM	62456.63441
ALO	62456.63309
GOA	**62441**
IGOA	**62137**

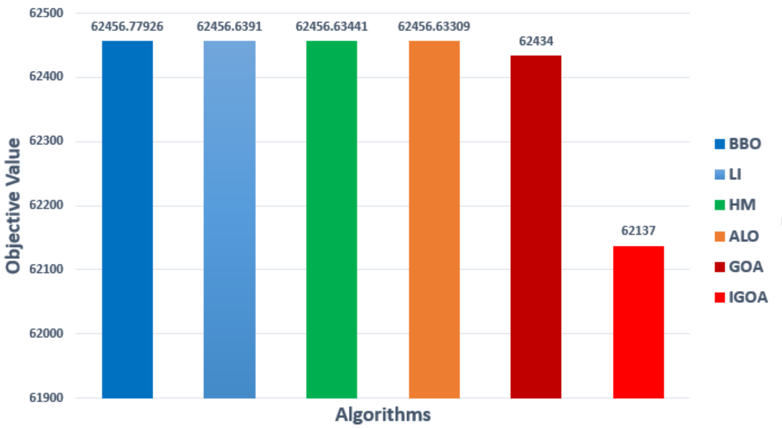

Figure 16. Best results obtained, for case study 4, by different techniques are compared with IGOA algorithm

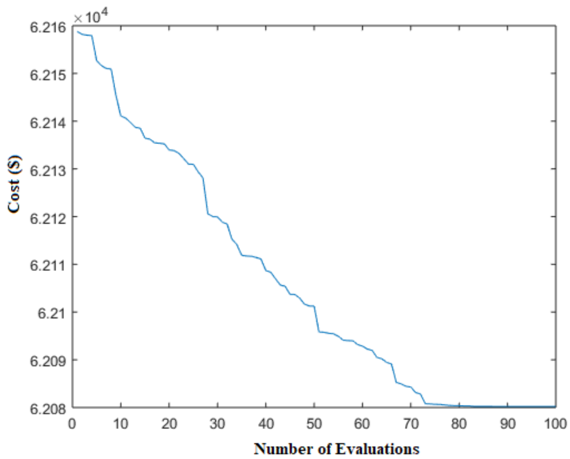

Figure 17. IGOA takes less than 100 function evaluations by getting better results for case study 4.

5. Discussion on results

In case study 1, the operating cost obtained by acting GA method is 15459.0$, PSO method is 15450.0$ ESO method is 15450.0$, DE method is 15450.0$, HS method is 15449.0$ and HHS method is 15449.0$. However, the proposed method IGOA minimize the cost up to 15393.92$.

In case study 2, the obtained output of LI method is 30359.3$, FFA method is 30334.0$, ALO method is 30333.9858$. However, proposed method reduce the cost up to 30243$. we have obtained best minimized cost with required power demand.

In case study 3, the obtained output from PSO method is 41896.66$, LI method is 41959.0$, FFA method is 41896.9$ and ALO method is 41896.6286$. However, the proposed method achieves minimum cost operating cost that is 41865$.

In case study 4, the generation cost obtained through BBO method is 62456.77926$, LI method is 62456.6391$, HM method is 62456.63441$ and ALO method is 62456.63309$. However, proposed method reduce the cost up to 62137$.

6. Conclusion

In this paper, we have proposed a new initialization approach for GOA which is named as IGOA. ELD problems are considered for testing the new algorithm. To check the reliability of the IGOA we have taken case studies in ELD problems with nonlinear, complex quadratic functions, limits on the generation units, transmission line capacity, valve point effects, and different prohibited zones are taken into account to further increase the complexity of the problem. Our approach is easy to implement, for solving optimization problems of difficult landscapes. Therefore, we have tested our proposed technique on several case studies in ELD problems. IGOA has achieved the desirable computational efficiency, better results, and improved convergence rate. Moreover, IGOA can be examined by solving ELD problems with discrete domains.

References

[1] I. Aljarah , A.Z. AlaM, H. Faris, M.A. Hassonah, S. Mirjalili and H. Saadeh, *Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm*, Cognitive Computation, **21**,1-18, 2018.

[2] A. Bhattacharya and P.K. Chattopadhyay, *Biogeography-based optimization for different economic load dispatch problems*, IEEE transactions on power systems, **25**, 1064-1077, 2010.

[3] A.H. Bindu and M.D. Reddy, *Economic load dispatch using cuckoo search algorithm*, Int. Journal Of Engineering Research and Applications, **3**, 498-502, 2013.

[4] M. Basu, *A simulated annealing-based goal-attainment method for economic emission load dispatch of fixed head hydrothermal power systems*, International Journal of Electrical Power & Energy Systems, **27**, 147-153, 2005.

[5] A. Bhattacharya and P.K. Chattopadhyay, *Solving complex economic load dispatch problems using biogeography-based optimization*, Expert Systems with Applications, **37**, pp.3605-3615, 2010.

[6] P.H. Chen and H.C. Chang, *Large-scale economic dispatch by genetic algorithm*, IEEE transactions on power systems, **10**, 1919-1926, 1995.

[7] H.M. Dubey, M. Pandit, B.K. Panigrahi and M. Udgir, *Economic load dispatch by hybrid swarm intelligence based gravitational search algorithm*, International Journal of Intelligent Systems and Applications, **5**, 21, 2013.

[8] M. Fesanghary, and M.M. Ardehali, *A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem*, Energy, **34**, 757-766, 2009.

[9] Z.W. Geem, J.H. Kim and G.V. Loganathan, *A new heuristic optimization algorithm: harmony search. simulation*, **76**, 60-68, 2001.

[10] Z.L. Gaing, *Particle swarm optimization to solving the economic dispatch considering the generator constraints*, IEEE transactions on power systems. **18**, 1187-95, 2003.

[11] Y.C. Ho and Q.C. Zhao and D.L. Pepyne, *The no free lunch theorems: Complexity and security*, IEEE Transactions on Automatic Control, **48**, 83-793, 2003.

[12] S. Hemamalini and S.P. Simon, *Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions*, Electric Power Components and Systems, **38**, 786-803, 2010.

[13] A. Lewis, *LoCost: a spatial social network algorithm for multi-objective optimisation*, In Evolutionary Computation, CEC'09, IEEE Congress, 2866-2870, 2009.

[14] S.Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris and I. Aljarah, *Grasshopper optimization algorithm for multi-objective optimization problems*, Applied Intelligence, **48**, 805-820, 2018.

[15] M.M. Nischal and S. Mehta, *Optimal load dispatch using ant lion optimization*, Int. J Eng Res Appl, **5**, 10-19, 2015.

[16] N. Noman and H. Iba, *Differential evolution for economic load dispatch problems.* Electric Power Systems Research, **78**, 1322-1331, 2008.

[17] R.E. Perez-Guerrero and J.R. Cedeno-Maldonado, *Economic power dispatch with non-smooth cost functions using differential evolution*, In Power Symposium, Proceedings of the 37th Annual North American, 183-190, 2005.

[18] V.R. Pandi, B.K. Panigrahi, R.C. Bansal, S. Das and A. Mohapatra, *Economic load dispatch using hybrid swarm intelligence based harmony search algorithm*, Electric power components and systems, **39**, 751-767, 2011.

[19] A. Pereira-Neto, C. Unsihuay, and O.R. Saavedra, *Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints*, IEE Proceedings-Generation, Transmission and Distribution, **152**, 653-660, 2005.

[20] S. Pothiya, I. Ngamroo and W. Kongprawechnon, *Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints*, Energy Conversion and Management. **49**, 506-16, 2008.

[21] S.R. Rayapudi, *An intelligent water drop algorithm for solving economic load dispatch problem*, International Journal of Electrical and Electronics Engineering, **5**, 43-49, 2011.

[22] K.S. Reddy, and M.D. Reddy, *Economic load dispatch using firefly algorithm*, International journal of Engineering Research and Applications, **2**, 2325-2330, 2012.

[23] S.M. Rogers, T. Matheson, E. Despland, T. Dodgson, M. Burrows and S.J. Simpson, *Mechanosensory-induced behavioural gregarization in the desert locust Schistocerca gregaria*, Journal of Experimental Biology, **206**, 3991-4002, 2003.

[24] S. Saremi, S. Mirjalili and A. Lewis, *Grasshopper optimisation algorithm: theory and application*, Advances in Engineering Software, **105**, 30-47, 2017.

[25] C.T. Su, and C.T. Lin, *New approach with a Hopfield modeling framework to economic dispatch*, IEEE Transactions on Power Systems, **15**, 541-545, 2000.

[26] N. Sinha, R. Chakrabarti and Chattopadhyay P.K. *Evolutionary programming techniques for economic load dispatch*, IEEE Transactions on Evolutionary Computation. **7**, 83-94, 2003.

[27] S. Sayah and A. Hamouda, *A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems.* Applied Soft Computing, **13**, 1608-1619, 2013.

[28] R.K. Swain, N.C. Sahu and P.K. Hota, *Gravitational search algorithm for optimal economic dispatch*, Procedia technology, **6**, 411-419, 2012.

[29] C.M. Topaz, A.J. Bernoff, S. Logan and W. Toolson, *A model for rolling swarms of locusts*, The European Physical Journal Special Topics, **157**, 93-109, 2008.

[30] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su and Y. Yu, *Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by Adaptive Grasshopper Optimization Algorithm*, Aerospace Science and Technology, **70**, 497-510, 2017.

[31] L. Wang and L.P. Li , *An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems*, International Journal of Electrical Power & Energy Systems, **44**, 832-843, 2013.

[32] A.J. Wood, and F.W. Bruce, *Power generation, operation, and control*, John Wiley and Sons, 2012.

[33] H.K. Youssef and K.M. El-Naggar, *Genetic based algorithm for security constrained power system economic dispatch*, Electric Power Systems Research, **53**, 47-51, 2000.

[34] X.S. Yang, S.S.S. Hosseini and A.H. Gandomi, *Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect*, Applied Soft Computing, **12**, 1180-1186, 2012.

[35] T. Yalcinoz, H. Altun, and M. Uzam, *Economic dispatch solution using a genetic algorithm based on arithmetic crossover*, In Power Tech Proceedings, IEEE Porto, **2**, 4-pp, 2001.