Al-Aqsa University Journal (Natural Sciences Series),Vol.22, Pages 1-35,Jan.2020
ISSN 2070-3155

HABC: Hybridizing artificial bee colony with *β*-hill climbing optimizer for solving non-convex economic load dispatch problem

Dr. Mohammed A. Awadallah*

الملخص

تهجين خوارزمية مستعمرة النحل الذكية مع خوارزمية بيتا-تسلق التل لحل مشكلة التوزيع الاقتصادي للأحمال

في أنظمة الطاقة، تعالج مشكلة التوزيع الاقتصادي للأحمال (ELD) بواسطة إعادة جدولة الطاقة الناتجة من وحدات التوليد لتقليل تكلفة الوقود المستهلكة. حيث أن هناك العديد من خوارزميات النمذجة المقترحة لحل هذه المشكلة. في هذا البحث، يتم اقتراح خوارزمية مستعمرة النحل الذكية الهجينة (HABC) لحل هذه المشكلة. تجدر الإشارة أن خوارزمية مستعمرة النحل تعتبر من خوارزميات النمذجة وهي تعاني من مشكلة رئيسية في الاستغلال سواءً في النحل العامل (Employed bees) أو النحل المتفرج (Onlooker bees). في الخوارزمية المقترحة HABC، تم استبدال وظيفة النحل المتفرج باستخدام النسخة المطورة من خوارزمية تسلق التل (β-hill climbing algorithm) وذلك من اجل تعزيز خاصية الاستغلال في الخوارزمية المقترحة HABC. تم دراسة تأثير إعدادات المتغيرات المختلفة علي سلوك الخوارزمية المقترحة باستخدام 9 سيناريوهات تجريبية مختلفة. استخدمت الخوارزمية المقترحة لحل 3 حالات مختلفة لمشكلة ELD، اثنتان مكونة من 13 مولد وواحدة مكونة من 40 مولد. وضحت النتائج أن الخوارزمية المقترحة حصلت علي ثاني أفضل النتائج بالمقارنة بالخوارزمية الأخرى في الثلاث حالات المختلفة لمشكلة ELD.

* قسم علوم الحاسوب – جامعة الاقصى- غزة- فلسطين.

Abstract

In power systems, economic load dispatch (ELD) problem is tackled by rescheduling the power outcomes of the generation units to minimized the fuel cost consumption. ELD is formulated as an optimization problem which is tackled by several optimization methods. In this paper, ELD is tackled by a hybrid artificial bee colony (HABC). Artificial bee colony, an efficient optimization method, has a chronic shortcoming in improvisation equation of employed and onlooker bees operators. In HABC, the onlooker bee operator is replaced by the β-hill climbing optimizer as new operator to empower its exploitation capability. HABC is evaluated using two different ELD problems with thirteen generating units, and one problem with forty generating units. The effect of the different parameter settings on the behavior of HABC is tested using nine experimental scenarios. The experimental results demonstrate that HABC is able to achieve the second best results for the three ELD problems.

1. Introduction

Recently, economic load dispatch (ELD) problem aroused the attention of research communities in the power system. In ELD, the main objective is to achieve the predefined power output obtained from the active generating units in the system with minimum fuel costs. This is done in accordance with satisfying equality and inequality constraints. The power balance is represented as the quality constraints, while the power output is represented as the inequality constraints. In the previous era, the traditional calculus-based optimization methods are proposed to solve ELD problems. Examples include gradient-based method (Dodu et al., 1972); linear programming algorithm (Jabr et al., 2000); non-linear programming algorithm (Nanda et al., 1994); quadratic programming (Coelho and Mariani, 2006); and lagrangian relaxation algorithm (El-Keib et al., 1994). Indeed, this method can be very efficient in solving the problem with low dimensionality. However, in large-dimensional ELD instances, this kind of method is impractical due to the fact that ELD is classified as a non-convex and highly non-linear optimization problem, and thus cannot lend itself to be solved easily by the traditional methods.

Metaheuristic-based method is the most efficient approach proposed for the ELD problems. This is because the method have the ability to solve complex problems with reasonable computational time (Blum and Roli, 2003). The common capability of any metaheuristic-based method resides in its power in exploring the unvisited region of the search space (exploration) and exploiting the accumulative search (exploitation). Balance between the exploration and the exploitation is the key success of seeking for the optimal solution. Conventionally, metaheuristic-based method could be classified into two main categories: local search-based, and population-based methods (Hussain et al., 2018).

Local search methods are initiated with one random solution. This solution is iteratively changed based on neighboring search process to come up with a new neighboring solution. The neighboring solution replaces the current one, if better, and this process is terminated as the local minima is achieved. Note that the exploitation is bias feature of the local-search based methods. There are many local search-based methods that have been used to solve ELD. This include: simulated annealing (Zhang et al., 2015), tabu search (Lin et al., 2002), GRASP (Neto et al., 2017), and β-hill climbing (Al-Betar et al., 2018).

Population-based algorithms are normally initiated with a set of solutions. The properties of these solutions are exchanged using learning processes controlled by specific extreme values until a (premature) optimal solution is reached. It is worthy of notice that the population-based methods is bias to towards exploration. In general, the population-based methods can be further subdivided into evolutionary computation-based methods (EC) and swarm intelligence-based methods (SI) (BoussaïD et al., 2013). EC algorithm is inspired by utilizing Darwin's principle of natural selection (i.e., survival of the fittest), whereas SI algorithm, which is like the one used in this study, is mostly inspired by the animal behaviors on seeking food or hunting process. Some of the examples of EC algorithm utilized for solving ELD problems are genetic algorithm (Shang et al., 2017); harmony search (dos Santos Coelho and Mariani, 2009); and evolutionary algorithm (Sinha et al., 2003), while the SI method employed for ELD problems include krill herd algorithm (Mandal et al., 2014), cuckoo search algorithm (Afzalan and Joorabian, 2015); bacterial foraging optimisation (Panigrahi and Pandi, 2008); firefly algorithm(Yang et al., 2012); ant colony algorithm (Pothiya et al., 2010); and particle swarm optimization (Selvakumar and Thanushkodi, 2007).

The Artificial Bee Colony (ABC) is a class of SI algorithm is proposed by Karaboga in 2005 (Karaboga, 2005). Due to its simplicity, flexibility, and robustness, ABC have been implemented to solve many optimization problems like nurse rostering problem (Awadallah et al., 2015), university timetabling (Bolaji et al., 2014), job-shop scheduling (Sundar et al., 2017), multi-threshold segmentation(Díaz-Cortés et al., 2017), knapsack problem(He et al., 2018), and others reported in (Akay and Karaboga, 2015, Bolaji et al., 2013, Karaboga et al., 2014). ABC suffers from shortcomings such as i) it easy to get stuck in local optima; ii) very slow when applied to solve hard problems, due to the high number of fitness evaluations at each iteration; and iii) the search equation of the employee and onlooker bees operators is poor in exploitation and good in exploration (Ab Wahab et al., 2015, Bolaji et al., 2013, Gao et al., 2012). Multiple versions of ABC have been proposed in order to bridge these shortcomings (Awadallah et al., 2015, Gao et al., 2012, He et al., 2018, Zhong et al., 2017).

In this paper, a hybrid version of ABC algorithm is proposed for solving the ELD problems, called HABC. In HABC, the following contributions are utilized:

- The β-hill climbing optimizer is replaced the functionality of the onlooker bee operator of the ABC algorithm in order to empower its exploitation capability, called HABC.
- The β-hill climbing rate (βHCR) is suggested as a control parameter to determine the percentage of using β-hill climbing optimizer in ABC algorithm.
- The HABC is experimentally evaluated using three non-convex ELD systems with diverse complexities and characteristics: two different ELD problems with thirteen generating units, and one problem with forty generating units.
- The sensitivity analysis step of HABC is studied to show the impact of its control parameters on its convergence behavior.
- The comparative evaluation for HABC concur that it achieves superior results when it compares with the available state of the art methods using the same ELD problems.

The paper is organized as follow: Section 2 presents the formulation of the ELD problem. Section 3 describes the different steps of the proposed HABC algorithm used to solve the ELD problem. Experimental results and analysis are provided in Section 4. Finally, conclusions and some future directions are summarized in Section 5.

2. ELD problem formulation

The Economic load dispatch (ELD) problem is defined as the process of allocating generation levels to be generated by each active generating units in the system. The main objective is to minimize the fuel cost of the generating units for a specific period of operations subject to satisfying the quality and inequality constraints. Generally, the ELD can be formulated as:

$$\text{Minimize} \quad \boldsymbol{F_T} = \sum_{i=1}^{N} F_i(P_i) \tag{1}$$

Where $\boldsymbol{F_T}$ is the total production cost in \$/hr; N is the total number of active generating units included in the system; and $F_i(P_i)$ is the fuel cost function for the generating unit i, which is calculated using the following equation:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + \left| e_i \sin\left(f_i\left(P_i^{min} - P_i\right)\right)\right| \tag{2}$$

Where a_i, b_i, and c_i are the smooth fuel cost coefficients of the generating unit i; e_i and f_i are the non-smooth fuel cost coefficients of the generating unit i; P_i is the electrical output power of generating unit i in MW; and $P_i^{min}$is the minimum generating limit of generating unit i.

The solution of the ELD problem is subjected to the following constraints:

1. Quality constraint (Power balance constraint):

$$\sum_{i=1}^{N} P_i = P_D + P_L \tag{3}$$

Where P_D is the total load demand in MW; P_L is the total transmission losses in the system in MW. It should be noted that the total transmission losses P_L is computed using B-coefficients as follows:

$$P_L = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} P_i B_{i0} + B_{00} \quad (4)$$

2. Inequality constraint (Power generation limits):

$$P_i^{min} \leq P_i \leq P_i^{max} \quad (5)$$

The output power (MW) of each generating unit i shall be within their minimum limit P_i^{min} and maximum limit P_i^{max}.

It is worth to mention that, the system transmission losses will be ignored for all the test cases considered in this research as the others (Al-Betar et al., 2018, Al-Betar et al., 2016a, Al-Betar et al., 2016b).

3. The proposed Method

In this section, the hybridization of ABC algorithm with β-hill climbing optimizer for tacking the non-convex ELD problems is presented.

The ABC algorithm is a SI algorithm introduced by Karaboga in 2005 (Karaboga, 2005). This algorithm simulates the intelligent foraging behavior of a honey bee colony. In general, the honey bees in the colony is divided into three groups based on foraging task: i) employed bees; ii) onlooker bees; and iii) scout bees. *Employed bees* are responsible to collect the nectars from the discovered food sources and transfer to the hive, as well as, dancing in the hive in order to share the information about the food sources with the onlooker bees in the hive. *Onlooker bees* are responsible to select one of the good food sources to exploit. Finally, the *scout bees* are responsible to discover new food sources randomly. In optimization terms, the employed and onlooker bees are the source of exploitation capability by exploiting the discovered food sources, while the scout bees is the source of the exploration capability by visiting new food source randomly. The procedural steps of the proposed HABC for ELD is illustrated in Figure 1, while the description of these steps are given below:

3.1 Initialize the parameters

In this step, the four parameters of the proposed HABC are initialized which includes:

- Solution Number (*SN*) represents the number of solutions (i.e., food sources) in the population.
- Maximum cycle number (*MCN*) reflects the maximum number of generations.
- *limit* represents the certain number of generations which is used to abandon the exhausted food source in the population.
- β-hill climbing rate (βHCR) refers to the rate of calling the β-hill climbing optimizer in order to enhance the desired food source, where its value between 0 and 1.

Similarly, the representation of the solutions, as well as, the cost function (see Eq. (1)) are initialized. Furthermore, the different parameters of the ELD problem, which are mentioned in Section 2, are extracted from the dataset.

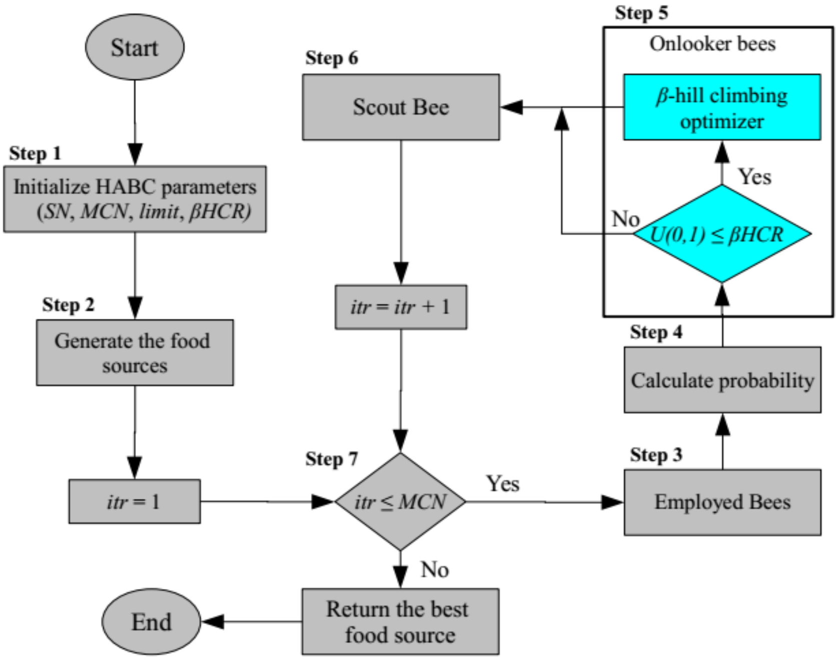

Figure 1: The flowchart of the proposed HABC algorithm

3.2 Generate the food sources

In this step, the initial solutions in the population are initialized, $\boldsymbol{BM}= [\boldsymbol{P_1}, \boldsymbol{P_2}, \ldots, \boldsymbol{P_{SN}}]^T$, where *SN* is the size of the population. Each solution $\boldsymbol{P_j}= \{P_{j1}, P_{j2}, P_{j3}, \ldots, P_{jN}\}$ in the population reflect one of the food sources, where *N* is represents the number of active generating units in the system. These initial solutions are randomly constructed as follows:

$$P_{ji} = P_i^{min} + (P_i^{max} - P_i^{min}) \times U(0,1) \qquad (6)$$

Where $j \in [1, SN]$ and $i \in [1, N]$, and $U(0,1)$ generates a random number between 0 and 1.

Then, simple repair procedure is triggered to ensure the feasibility of each solution by satisfying the quality and inequality constraints. In repair process, each generating unit P_{ji} is check to ensure the output power assigned is between $P_i^{min}$and P_i^{max}. If P_{ji} is less thanP_i^{min}or bigger thanP_i^{max}, then P_{ji} is assigned a random value between $P_i^{min}$and P_i^{max}. On other hand, if the summation of the output power for all generating unit in the system is not met the total load demand then these differences will be added or removed from the different generating units.

Finally, each solution in the population is subjected to calculate the total production cost using Eq. (1).

3.3 Sending employed bees for food sources

Every food source (i.e., solution) is under the responsibility of one employed bee, in which each employed bee modifies its current associated solution $\boldsymbol{P_j}$ to produces the neighborhood solution $\boldsymbol{P_j'}$ using Eq. (7).

$$P'_{ji} = P_{ji} + \varphi(P_{ji} + P_{ki}) \tag{7}$$

Where $\boldsymbol{P_j}$ is the current associated solution; $\boldsymbol{P_k}$ is other solution selected randomly, where k must different from j; i is the position of the generating unit to be perturbated; and φ is a random number between -1 and 1, that is used to move from the current solution $\boldsymbol{P_j}$ to new one $\boldsymbol{P_j'}$. It should be noted that if the total production cost of the new solution $\boldsymbol{P_j'}$ is less than that of its current associated solution $\boldsymbol{P_j}$, then the employed bee release the old one $\boldsymbol{P_j}$ and memorize the new solution $\boldsymbol{P_j'}$ as the current associated solution.

3.4 Calculation of probability for food sources

When the employed bees complete their search using Eq. (7), each employed bee share the information of the found food sources to the onlooker bees. The onlooker bee chooses a food source for further search depending on the probability. The nectar value in each food sources is used to calculate the probability. In this step, the probability of each solution in the population is assigned using Eq. (8), where this probability is calculated depending on the value of the total cost production (see Eq. (1)).

$$Probability(\boldsymbol{P}_j) = \frac{\boldsymbol{F_T}(\boldsymbol{P}_j)}{\sum_{r=1}^{SN} \boldsymbol{F_T}(\boldsymbol{P}_r)} \tag{8}$$

Where the $\boldsymbol{F_T}(\boldsymbol{P}_J)$ is the value of the total cost production of solution $\boldsymbol{P}_J$; SN is the population size; and $\sum_{r=1}^{SN} Probabitity(\boldsymbol{P}_r)$ is unity.

3.5 Sending onlooker bees to food sources

The onlooker bee chooses one of the fittest food sources for further search using the same search equation of the employed bees. Likewise, if the new solution is better than the old one, its replaces the old one in the population.

It should be noted that the search equation of employed and onlooker bees is good in exploration and poor in exploitation (Gao et al., 2012). For this reasons, the search equation of the onlooker bee is replaced using β-hill climbing optimizer in order to enhance the desired solution until maximum limit is achieved. Based on the above-mentioned shortcoming, the proposed method is known as HABC, a new hybrid version of artificial bee colony algorithm is proposed for solving the ELD problem.

β-hill climbing optimizer is a modified version of the simple local search known as hill climbing algorithm, it is proposed by Al-Betar in 2017 (Al-Betar, 2017). In this algorithm, a new intelligent operator (*β-operator*) is added to the body of the algorithm in order to escape being stuck in local optima. *β-operator* is similar to mutation operator in genetic algorithm, which is the main source of randomness.

In this phase, the desired solution is passed to the β-hill climbing optimizer in order to find the local optima. It should be noted that the β-hill climbing optimizer is triggered depending on β-hill climbing rate (βHCR). The higher value of βHCR leads to higher probability of running the β-hill climbing optimizer, and thus the higher the rate of exploitations and the higher the CPU time.

3.6 Scout bees phase

If there exist any solution in the population that is not enhanced for a given number of generations (as determined by *limit* parameter), the scout bee replaces this abandoned solution with a new one, which is generated randomly using Eq. (6).

3.7 Stop condition

Steps 3.3 to 3.6 are repeated until the maximum number of generations is reached (MCN).

4. Results and discussions

The proposed HABC algorithm is used to solve ELD problems using three different test cases in order to evaluate its performance. These test cases are i) 13 generating units with required load demand is 1800MW; ii) 13 generating units with required load demand is 2520MW; and iii) 40 generating units with required load demand is 10500MW.

4.1 Experimental Setup

The performance of the proposed HABC is tested using nine convergence scenarios are provided in Table 1. These scenarios are divided into three groups in order to study the three parameters of the proposed method. Firstly, the *SN* parameter is studied using three different values (i.e., *SN*=10, *SN*=20, and *SN*=30) in three different convergence scenarios (i.e., Sen1, Sen2, and Sen3). Whereas the value of the other parameters in these scenarios are fixed like *limit*=*SN*×*D* and *βHCR*=0.05. It should be noted that the value of the *SN* parameter that obtained the best results in Sen1 - Sen3 will be used in the following experimental scenarios.

Secondly, the next three convergence scenarios (Sen4 - Sen6) are designed to study the effect of the *limit* parameter using three values (i.e., *limit*=0.5×*SN*×*D*, *limit*=*SN*×*D*, and *limit*=2×*SN*×*D*). The value of the *βHCR* parameter in these scenarios is fixed to 0.05. Again the value of the *limit* parameter that achieved the best results in these scenarios will be used in the next experiments. Finally, the last three scenarios are designed to study of the effect *βHCR* parameter on the performance of the proposed algorithm.

It is worth of mentioning that the proposed HABC algorithm is implemented using MATLAB VersionR2014b. The implemented code is executed on Corei7machine with16 GB RAM and Microsoft windows 10as operating system.

Table 1: Experimental scenarios used to study the behavior the proposed HABC.

Scenario	*SN*	*Limit*	*βHCR*
Sen1	10	SN×D	0.05
Sen2	20		
Sen3	30		
Sen4		0.5×SN×D	0.05
Sen5		SN×D	

Sen6	2×SN×D
Sen7	0.005
Sen8	0.05
Sen9	0.5

4.2 Test System 1

This test system comprises of thirteen generating units with non-convex cost functions. The total load demand was assumed to be 1800MW. The generating unit data of this problem is collected form (Walters and Sheble, 1993).

The results of studying the *SN* parameter using three different values in three convergence scenarios are shown in Table 2. In this table, the best solution obtained over 25 runs for each convergence scenario is recorded. Furthermore, as shown in this table the total cost of the best solution, the mean of the results, as well as the standard derivations are summarized. The best result obtained is highlighted in **bold** font. From Table 2, it is obvious that the best result is obtained by *Sen3*, where the value of *SN* is the highest. This is because the higher value of *SN* leads the proposed method to cover a larger area of the problem search space and thus the probabilities to reach better results are increased. It should be noted that the value of *SN* is set to 30 in the next experiments. It worth of mentioning that if *SN* is higher than 30 there are no much difference in the obtained results which leads to undesirable computational time.

Table 2: Effect of *SN* parameter for Test System 1

Unit	*Sen1*	*Sen2*	*Sen3*
g_1	628.3185	628.3185	628.3185
g_2	224.3995	224.3993	149.5814
g_3	148.0042	147.9598	222.8007
g_4	109.8665	109.8658	109.8583
g_5	109.8662	109.8658	109.8522
g_6	60	109.8641	60
g_7	109.8289	109.8602	109.8567
g_8	109.8596	60	109.8657

g_9	109.8565	109.8665	109.8663
g_{10}	40	40	40
g_{11}	40	40	40
g_{12}	55	55	55
g_{13}	55	55	55
Total cost($)	17,960.67	17,960.54	**17,960.51**
Mean Cost	17,965.53	17,963.24	17,961.62
Stdev	3.84	3.28	1.43

The results of the three convergence scenarios (*Sen4* – *Sen6*) that are used to study the *limit* parameter on this problem are provided in Table 3. This table summarizes the cost of each generating unit in the best solution achieved, the total cost of the best solution, the mean of the results, and the standard derivations. Again, each convergence scenario is run 25 times. The cost of the best solution obtained is highlighted in **bold** font.

The recorded results in Table 3 demonstrate that the performance of the proposed HABC algorithm is affected by the varying value of *limit* parameter. As shown in this table, the best results are obtained by *Sen6*, where the value of *limit* parameter is the highest. Furthermore, the results obtained by the other scenarios (*Sen4* and *Sen5*) are very close to results of *Sen6*.

Finally, in order to show the effect of variation of the *βHCR* parameter on the performance of the proposed HABC method, three convergence scenarios are designed with three values of *βHCR* parameter (i.e., *Sen7* (*βHCR*=0.005), *Sen8* (*βHCR*=0.05), and *Sen9* (*βHCR*=0.5)). It should be noted that when the value of *βHCR* is higher, then it leads to the higher calling of *β*-hill climbing algorithm and thus increase the rate of exploitation and CPU time. Table 4 shows the comparison of the best solutions obtained by *Sen7* to *Sen9*. As seen in this table, the minimum total cost ($17,960.38)is obtained by *Sen8*, where the value of *βHCR* is set to 0.05. This is because the value of *βHCR* parameter achieved the considerable balance between the exploitation and exploration capabilities during the navigation of the search space of the problem. However, the performance of the *Sen7* is the worst, this is because the value of *βHCR* parameter is low and thus leads to high rate of exploration and low rate of exploitation.

Table 3: Effect of *limit* parameter for Test System 1

Unit	*Sen4*	*Sen5*	*Sen6*
g_1	628.3185	628.3185	628.3185

g_2	224.3993	149.5814	149.5987
g_3	147.9564	222.8007	222.7538
g_4	60	109.8583	109.8665
g_5	109.8658	109.8522	109.8665
g_6	109.8667	60	109.8655
g_7	109.8665	109.8567	60.0000
g_8	109.8644	109.8657	109.8665
g_9	109.8625	109.8663	109.8639
g_{10}	40	40	40
g_{11}	40	40	40
g_{12}	55	55	55
g_{13}	55	55	55
Total cost($)	17,960.53	17,960.51	**17,960.38**
Mean Cost	17,962.09	17,961.62	17,961.24
Stdev	1.70	1.43	0.55

Table 4: Effect of βHCR parameter for Test System 1

Unit	*Sen7*	*Sen8*	*Sen9*
g_1	628.3184	628.3185	628.3185
g_2	224.0833	149.5987	149.5908
g_3	148.5210	222.7538	222.7624
g_4	109.7881	109.8665	109.8661
g_5	109.8318	109.8665	109.8654
g_6	60	109.8655	109.8659
g_7	109.8413	60	109.8651
g_8	109.8025	109.8665	60
g_9	109.8135	109.8639	109.8657
g_{10}	40	40	40
g_{11}	40	40	40
g_{12}	55	55	55
g_{13}	55	55	55
Total cost($)	17,961.91	**17,960.38**	17,960.40
Mean Cost	17,987.79	17,961.24	17,960.92
Stdev	73.5498	0.55	1.36

Figure 2 shows the box plot for the experimental results of *Sen1* to *Sen9* based on the results recorded in Tables 2 – 4 on Test System 1. It can be seen from the figure that the distribution of the results obtained over 25 runs for each experimental scenario. The *x-axis* represent the proposed experimental scenarios, while *y-axis* represents the total fuel cost achieved. Clearly, *Sen9* is statistically better than the other proposed scenarios, where the distance between the best, median, and the worst results is the smallest.

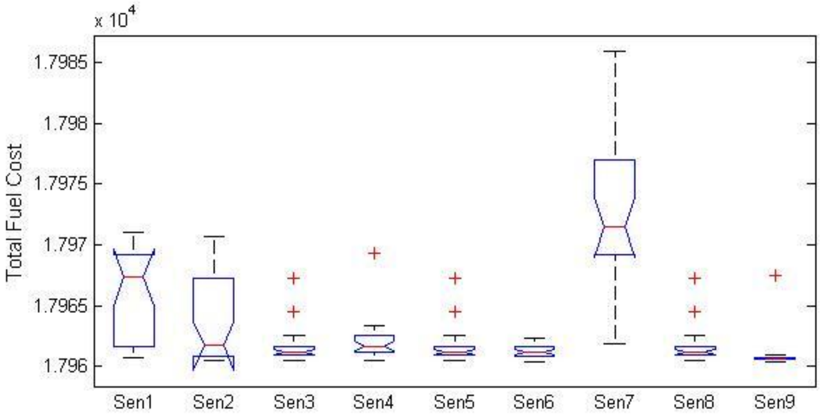

Figure 2: Box plot for *Sen1* – *Sen9* of the proposed HABC algorithm on Test System 1

For comparative evaluation purposes, the best result obtained by the proposed HABC algorithm is compared with those achieved by the other methods as shown in Table 5. The comparative methods include ABOMDE (Lohokare et al., 2012), FCASO-SQP (Cai et al., 2012b), GA-PS-SQP (Alsumait et al., 2010), HCASO (Cai et al., 2012b), HHS (Pandi et al., 2011), HMAPSO (Kumar et al., 2011), HQIPSO (Chakraborty et al., 2011), HS (dos Santos Coelho and Mariani, 2009), HIS (dos Santos Coelho and Mariani, 2009), QIPSO (Azizipanah-Abarghooee et al., 2012), NUHS (Al-Betar et al., 2016b), and THS (Al-Betar et al., 2016a). As shown in this table, the proposed HABC algorithm outperforms ten out of 14 comparative methods. However, the best results ($17,960.37) is achieved by four comparative methods, while the proposed algorithm obtained the second best results ($17,960.38).This is prove that the proposed algorithm can be able to make a right balance between the exploration and exploitation while navigating the problem search space and thus achieve better results. Figure 3 compares the minimum fuel costs obtained by the comparative methods.

Table 5: Comparison results of Test System 1

Method	Best	Mean
HABC	17,960.38	17,961.24
ABOMDE	17,963.85	17,967.36
FCASO-SQP	17,964.08	18,001.96
GA-PS-SQP	17,964.00	18,199.00
HCASO	17,965.15	18,022.04
HHS	17,963.83	17,972.48
HMAPSO	17,969.31	17,969.31
HQIPSO	17,966.37	18,081.05
HS	17,965.62	17,986.56
HIS	**17,960.37**	17,965.42
QIPSO	17,969.01	18,075.11
NUHS	**17,960.37**	17,987.10
THS	**17,960.37**	17,977.60

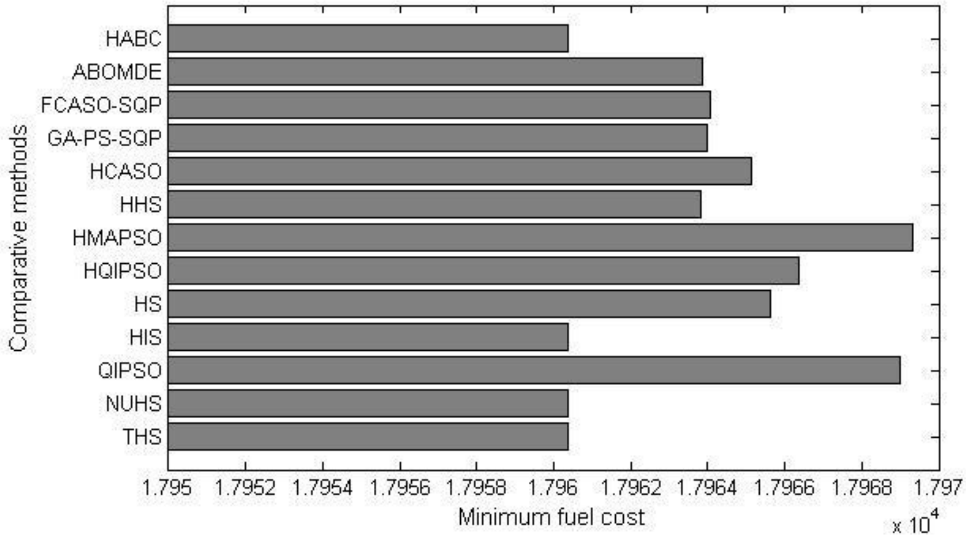

Figure 3: Minimum fuel cost comparison for Test System 1

4.3 Test System 2

This system considers thirteen generating units with a load demand of 2520MW. The problem instance is reported in (Walters and Sheble, 1993). Tables 6, 7, and 8 summarize the results of studying the performance of the proposed method using nine convergence scenarios as provided in Table 1. The tables 6, 7, and 8 recorded the best solution obtained by each convergence scenario, the cost of this solution, the mean of the results over 25 runs, and the standard derivation. The best result achieved is highlighted using **bold** font.

Table 6 shows the results of *Sen1* to *Sen3* that are employed to study the effect of *SN* parameter using three different values (i.e., *SN*=10, *SN*=20, and *SN*=30). Apparently, the performance of the three scenarios are almost similar, where the difference on the total cost less than or equal to 0.04. However, *Sen3* where the *SN* parameter is 30 achieved best result, and this value will be use in the next experimental scenarios.

Similarly, the results of studying the performance of *Sen4* to *Sen6* on Test System 2 are shown in Table 7. Again, these scenarios are designed in order to study the effect of *limit* parameter using varying values. The results in Table 7 clearly show the effectiveness of the *limit* parameter on the performance of the proposed method, whereas *Sen6* is successfully achieved the best solution. This is proven that the $SN \times D$ is suitable threshold value of the *limit* parameter to diversify the population, and this value is used in the next phase of experiments.

Table 6: Effect of *SN* parameter for Test System 2

Unit	*Sen1*	*Sen2*	*Sen3*
g_1	628.3185	628.3185	628.3185
g_2	299.1938	299.1992	299.1993
g_3	294.5198	294.5203	294.4938
g_4	159.7331	159.7328	159.7318
g_5	159.7326	159.7329	159.7331
g_6	159.7326	159.7331	159.7330
g_7	159.7328	159.7330	159.7306

g_8	159.7329	159.7331	159.7329
g_9	159.7329	159.7330	159.7323
g_{10}	77.3895	77.3771	77.3949
g_{11}	77.3852	77.3999	77.3999
g_{12}	92.4006	92.4001	92.4000
g_{13}	92.3959	92.3870	92.4000
Total cost($)	24,164.12	24,164.11	**24,164.08**
Mean Cost	24,167.92	24,165.44	24,164.89
Stdev	7.21	1.20	0.97

Table 7: Effect of *limit* parameter for Test System 2

Unit	*Sen4*	*Sen5*	*Sen6*
g_1	628.3185	628.3185	628.3185
g_2	299.2000	299.1993	299.1991
g_3	294.5113	294.4938	294.5094
g_4	159.7253	159.7318	159.7331
g_5	159.7327	159.7331	159.7313
g_6	159.7266	159.7330	159.7325
g_7	159.7327	159.7306	159.7331
g_8	159.7331	159.7329	159.7331
g_9	159.7308	159.7323	159.7328
g_{10}	77.3990	77.3949	77.3988
g_{11}	77.3990	77.3999	77.3999
g_{12}	92.3904	92.4000	92.3787
g_{13}	92.4005	92.4000	92.3996
Total cost($)	24,164.14	**24,164.08**	24,164.09
Mean cost	24,164.74	24,164.89	24,164.95
Stdev	0.46	0.97	1.10

The best solutions achieved by studying the *βHCR* parameter using three different values are summarized in Table 8. Clearly, *Sen8* succeed to obtain the minimum total fuel cost ($24,164.08). This is proven that the 0.05 is the threshold value of the *βHCR* parameter and is able to make the right balance between the exploration and exploitation.

Table 8: Effect of βHCR parameter for Test System 2

Unit	*Sen7*	*Sen8*	*Sen9*
g_1	628.3185	628.3185	628.3185
g_2	299.1880	299.1993	299.1992
g_3	294.7467	294.4938	294.4990
g_4	159.7324	159.7318	159.7331
g_5	159.5931	159.7331	159.7330
g_6	159.7339	159.7330	159.7329
g_7	159.7276	159.7306	159.7328
g_8	159.7258	159.7329	159.7285
g_9	159.7324	159.7323	159.7295
g_{10}	77.3829	77.3949	77.3971
g_{11}	77.3880	77.3999	77.3970
g_{12}	92.3743	92.4000	92.3998
g_{13}	92.3563	92.4000	92.3996
Total cost($)	24,164.69	**24,164.08**	24,164.09
Mean cost	24,172.47	24,164.89	24,164.36
Stdev	7.83	0.97	0.17

Figure 4 shows the box that illustrates the distribution of the results of *Sen1* to *Sen9* on Test System 2. It can be seen from the figure that *Sen9* is statistically better than the other proposed scenarios.

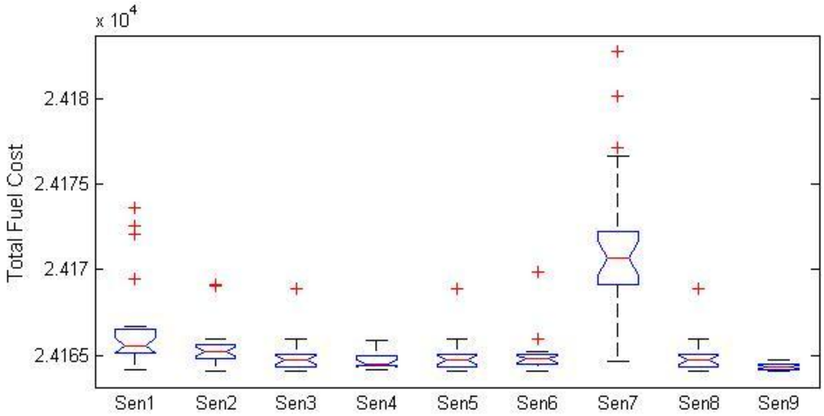

Figure 4: Box plot for *Sen1* – *Sen9* of the proposed HABC algorithm on Test System 2

The best result obtained by the proposed HABC against those obtained by the competitors are recorded in Table 9. In this table, the best results obtained by the comparative methods as well as the mean of the results are recorded. The best results obtained are highlighted using **bold** font. The comparative methods include ACO (Pothiya et al., 2010), FCASO-SQP (Cai et al., 2012b), HCASO (Cai et al., 2012a), HCPSO (Cai et al., 2012a), HCPSO-SQP (Cai et al., 2012a), TS (Pothiya et al., 2010), TSA (Khamsawang and Jiriwibhakorn, 2010), NUHS (Al-Betar et al., 2016b), THS (Al-Betar et al., 2016a), and IGWO (Mehmood and Ahmad, 2017). It can be observed from Table 9 that the performance of HABC is significantly better than eight out of ten comparative methods from the literature. Clearly, the best results ($24,164.06) are obtained by two of the comparative methods, while the proposed HABC is ranked second ($24,164.08). This is clearly shows the potential of the proposed method. Figure 5 illustrates the comparison of the results of the comparative methods in term of the minimum fuel cost.

Table 9: Comparison results of Test System 2

Method	Best	Mean
HABC	24,164.08	24,164.89
ACO	24,174.39	24,211.09
FCASO-SQP	24,190.63	NA
HCASO	24,212.93	NA
HCPSO	24,211.56	NA
HCPSO-SQP	24,190.97	NA
TS	24,180.31	24,243.37
TSA	24,171.21	24,184.06
NUHS	**24,164.06**	24,185.61
THS	**24,164.06**	24,195.21
IGWO	24,202.26	24,210.00

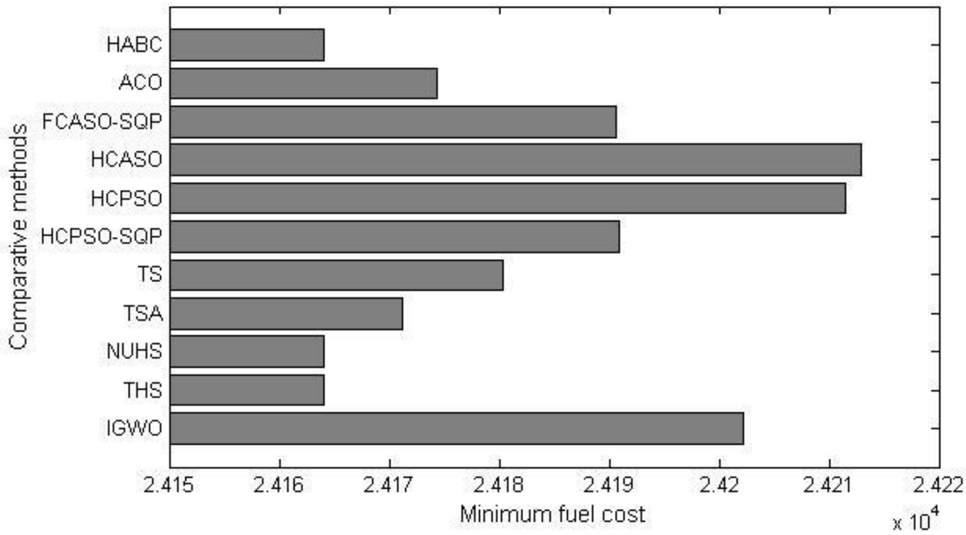

Figure 5: Minimum fuel cost comparison for Test System 2

4.4 Test System 3

In order to evaluate the efficiency of the proposed method using larger dataset, a problem instance with forty generating units is employed. The total expected load demand is 10500MW. The system parameters are taken from (Sinha et al., 2003). It should be noted that, the nine convergence scenarios provided in Table 1 are studied for this test system, and the experimental results are summarized in Tables 10, 11, and 12. The best results achieved are highlighted using **bold** numbers. Again, each convergence scenario is repeated 25 independent runs.

The results of studying the behavior of the proposed HABC using various values of *SN* parameter are recorded in Table 10. It can be seen that *Sen1* and *Sen2* obtained the same best total cost ($121,414.64). However, the result of *Sen3* is very close to results of the other scenarios with higher computational time. Based on above, the value of the *SN* parameter will be set to 10 in the next experiments.

Table 10: Effect of *SN* parameter for Test System 3

Unit	*Sen1*	*Sen2*	*Sen3*
g_1	110.7998	110.7999	110.7998
g_2	110.8008	110.8008	110.8003
g_3	97.3999	97.3999	97.3999
g_4	179.7331	179.7331	179.7331
g_5	92.7250	87.8292	92.6843
g_6	140	140	140
g_7	259.5997	259.5997	259.5997
g_8	284.5997	284.5997	284.5997
g_9	284.5997	284.5997	284.5997
g_{10}	130	130	130
g_{11}	168.7998	168.7999	168.7998
g_{12}	168.7998	168.7999	168.7998
g_{13}	214.7598	214.7598	214.7598
g_{14}	394.2797	394.2794	394.2794
g_{15}	394.2794	394.2793	394.2794
g_{16}	304.5196	304.5196	304.5196
g_{17}	489.2794	489.2794	489.2794
g_{18}	489.2794	489.2794	489.2794
g_{19}	511.2794	511.2794	511.2794
g_{20}	511.2794	511.2794	511.2794
g_{21}	523.2794	523.2794	523.2794
g_{22}	523.2794	523.2794	523.2794

g_{23}	523.2794	523.2794	523.2794
g_{24}	523.2794	523.2794	523.2794
g_{25}	523.2794	523.2794	523.2794
g_{26}	523.2794	523.2794	523.2794
g_{27}	10	10	10
g_{28}	10	10	10
g_{29}	10	10	10
g_{30}	87.8311	92.7277	87.8733
g_{31}	190	190	190
g_{32}	190	190	190
g_{33}	190	190	190
g_{34}	164.7999	164.7998	164.7998
g_{35}	164.7999	164.7999	164.7998
g_{36}	164.8002	164.7998	164.7998
g_{37}	110	110	110
g_{38}	110	110	110
g_{39}	110	110	110
g_{40}	511.2794	511.2794	511.2794
Total cost($)	**121,414.64**	**121,414.64**	121,414.66
Mean Cost	121,453.26	121,438.58	121,430.10
Stdev	26.23	22.26	21.99

The results obtained by *Sen4* to *Sen6* that are used to study the behavior of the proposed method using various values of *limit* parameter as reported in Table 11. It is observed from the results summarized in this table, *Sen6* obtained the best solution, while *Sen4* achieved the worst results. Which proved that the lower value of *limit* parameter leads to undesirable diversify of the population and thus achieved worst results. The value of $2 \times SN \times D$ is set for *limit* parameter in the next experimental scenarios.

Table 11: Effect of *limit* parameter for Test System 3

Unit	*Sen4*	*Sen5*	*Sen6*
g_1	110.8013	110.7998	110.8000
g_2	110.8017	110.8008	110.8005
g_3	97.4000	97.3999	97.3999
g_4	179.7331	179.7331	179.7331

g_5	92.6424	92.7250	87.8213
g_6	140	140	140
g_7	259.5997	259.5997	259.5997
g_8	284.5998	284.5997	284.5997
g_9	284.5998	284.5997	284.5997
g_{10}	130	130	130
g_{11}	168.7998	168.7998	168.7998
g_{12}	168.7998	168.7998	168.7998
g_{13}	214.7598	214.7598	214.7598
g_{14}	394.2795	394.2797	394.2794
g_{15}	304.5196	394.2794	304.5196
g_{16}	394.2794	304.5196	394.2794
g_{17}	489.2794	489.2794	489.2794
g_{18}	489.2794	489.2794	489.2794
g_{19}	511.2793	511.2794	511.2794
g_{20}	511.2794	511.2794	511.2794
g_{21}	523.2794	523.2794	523.2794
g_{22}	523.2794	523.2794	523.2794
g_{23}	523.2794	523.2794	523.2794
g_{24}	523.2794	523.2794	523.2794
g_{25}	523.2794	523.2794	523.2794
g_{26}	523.2794	523.2794	523.2794
g_{27}	10	10	10
g_{28}	10	10	10
g_{29}	10	10	10
g_{30}	87.9115	87.8311	92.7358
g_{31}	190	190	190
g_{32}	190	190	190
g_{33}	190	190	190
g_{34}	164.7998	164.7999	164.7998
g_{35}	164.8000	164.7999	164.7999
g_{36}	164.7999	164.8002	164.7999
g_{37}	110	110	110
g_{38}	110	110	110
g_{39}	110	110	110
g_{40}	511.2794	511.2794	511.2794
Total cost($)	121,414.68	121,414.64	**121,414.63**
Mean Cost	121,421.19	121,453.26	121,427.69

Stdev	11.93	26.23	19.71

Similarly, the results of studying the behavior of the proposed HABC method using three different values of *βHCR* parameter are provided in Table 12. Apparently, the performance of the proposed method is improved as the value of *βHCR* increased. The performance of *Sen7* is the worst, this is because the value of *βHCR* is the lowest. The best result is obtained by *Sen8*, when the value of *βHCR* is set to 0.05. However, when the value of *βHCR* increased to 0.5 by*Sen9*, the performance of the proposed method is more stable, but with results worst than *Sen8*. This is because the higher value of *βHCR* leads to fast convergence and thus achieved worst results.

Table 12: Effect of *βHCR* parameter for Test System 3

Unit	*Sen7*	*Sen8*	*Sen9*
g_1	110.8004	110.8000	110.7999
g_2	110.8082	110.8005	110.7999
g_3	97.4000	97.3999	97.3999
g_4	179.7331	179.7331	179.7331
g_5	87.9263	87.8213	92.7143
g_6	140	140	140
g_7	259.5997	259.5997	259.5997
g_8	284.5997	284.5997	284.5997
g_9	284.5997	284.5997	284.5997
g_{10}	130	130	130
g_{11}	168.8000	168.7998	168.7998
g_{12}	168.7998	168.7998	168.7998
g_{13}	214.7598	214.7598	214.7598
g_{14}	394.2794	394.2794	394.2794
g_{15}	394.2794	304.5196	394.2794
g_{16}	304.5197	394.2794	304.5196
g_{17}	489.2794	489.2794	489.2794
g_{18}	489.2794	489.2794	489.2794
g_{19}	511.2794	511.2794	511.2794
g_{20}	511.2794	511.2794	511.2794
g_{21}	523.2794	523.2794	523.2794
g_{22}	523.2794	523.2794	523.2794
g_{23}	523.2795	523.2794	523.2794
g_{24}	523.2794	523.2794	523.2794
g_{25}	523.2794	523.2794	523.2794

g_{26}	523.2794	523.2794	523.2794
g_{27}	10	10	10
g_{28}	10	10	10
g_{29}	10	10	10
g_{30}	92.6220	92.7358	87.8436
g_{31}	190	190	190
g_{32}	190	190	190
g_{33}	190	190	190
g_{34}	164.7998	164.7998	164.7998
g_{35}	164.7999	164.7999	164.7998
g_{36}	164.7999	164.7999	164.7998
g_{37}	110	110	110
g_{38}	110	110	110
g_{39}	110	110	110
g_{40}	511.2794	511.2794	511.2794
Total cost($)	121,414.70	**121,414.63**	121,414.64
Mean Cost	121,477.67	121,427.69	121,426.41
Stdev	36.72	19.71	17.55

Figure 6 illustrates the box plot of the results of *Sen1* to *Sen9* recorded in Tables 10 - 12. It can be seen that *Sen4* and *Sen9*are statistically better than the other proposed scenarios. This is indicate that the proposed HABC with the parameter settings that are used in these scenarios are the best to make the HABC more stabile able solve this case of ELD problem.

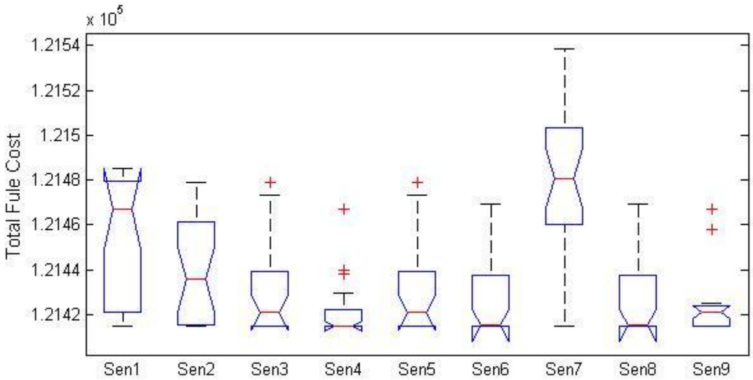

Figure 6: Box plot for *Sen1 – Sen9* of the proposed HABC algorithm on Test System 3

In order to demonstrate the strength of the proposed HABC method when compared with the other methods from the literature using a large-scaled and highly complex real-world dataset. The best result obtained by the proposed method as well as those achieved by the others are summarized in Table 13. In this table, the best results as well as the mean of the results are recorded. It should be noted the best recorded results are highlighted using **bold** font. The comparative methods include 26 algorithms such as ABOMDE (Lohokare et al., 2012), ACO (Pothiya et al., 2010), ARCGA (Sayah and Hamouda, 2013), BGO (Bhattacharya and Chattopadhyay, 2010b), CBPSO-RVM (Lu et al., 2010), CSOMA (dos Santos Coelho and Mariani, 2010), DE-BGO(Bhattacharya and Chattopadhyay, 2010a), FAPSO (Niknam et al., 2011), FAPSO-NM (Niknam et al., 2011), FCASO-SQP (Cai et al., 2012b), FFA (Yang et al., 2012), GA-PS-SQP (Alsumait et al., 2010), GSO (Moradi-Dalvand et al., 2012), HCPSO-SQP (Cai et al., 2012a), HHS (Pandi et al., 2011), HMAPSO (Kumar et al., 2011), HQIPSO (Chakraborty et al., 2011), NDS (Lin et al., 2011), QIPSO (Meng et al., 2010), TLA (Azizipanah-Abarghooee et al., 2012), TS (Pothiya et al., 2010), TSARGA (Subbaraj et al., 2011), NUHS (Al-Betar et al., 2016b), and THS (Al-Betar et al., 2016a). Interestingly, the proposed method obtained the second best results ($121,414.63), while the best results ($121,412.74) get by NUHS algorithm. This is proven that the proposed method can be used efficiently to solve highly complex cases of ELD problem. Figure 7 compares the minimum fuel cost of the comparative methods.

Table 13: Comparison results of Test System 3

Method	Best	Mean
HABC	121,414.63	121,427.69
ABOMDE	121,414.87	121,487.85
ACO	121,811.37	121,930.58
ARCGA	121,415.50	121,462.15
BGO	121,479.50	121,512.06
CBPSO-RVM	121,555.32	122,281.14
CSOMA	121,414.70	121,415.05
DE-BGO	121,420.89	121,420.90
FAPSO	121,712.40	121,778.25

FAPSO-NM	121,418.30	121,418.80
FCASO-SQP	121,456.98	122,026.21
FFA	121,415.05	121,416.57
GA-PS-SQP	121,458.00	122,039.00
GSO	124,265.40	124,609.18
HCPSO-SQP	121,458.54	122,028.16
HHS	121,415.59	121,615.85
HMAPSO	121,586.90	121,586.90
HQIPSO	121,418.60	121,427.47
NDS	121,647.40	121,647.40
QIPSO	121,448.21	122,225.07
TLA	122,009.77	122,074.90
TS	122,288.38	122,424.81
TSARGA	121,463.07	122,928.31
NUHS	**121,412.74**	121,549.95
THS	121,425.15	121,528.65

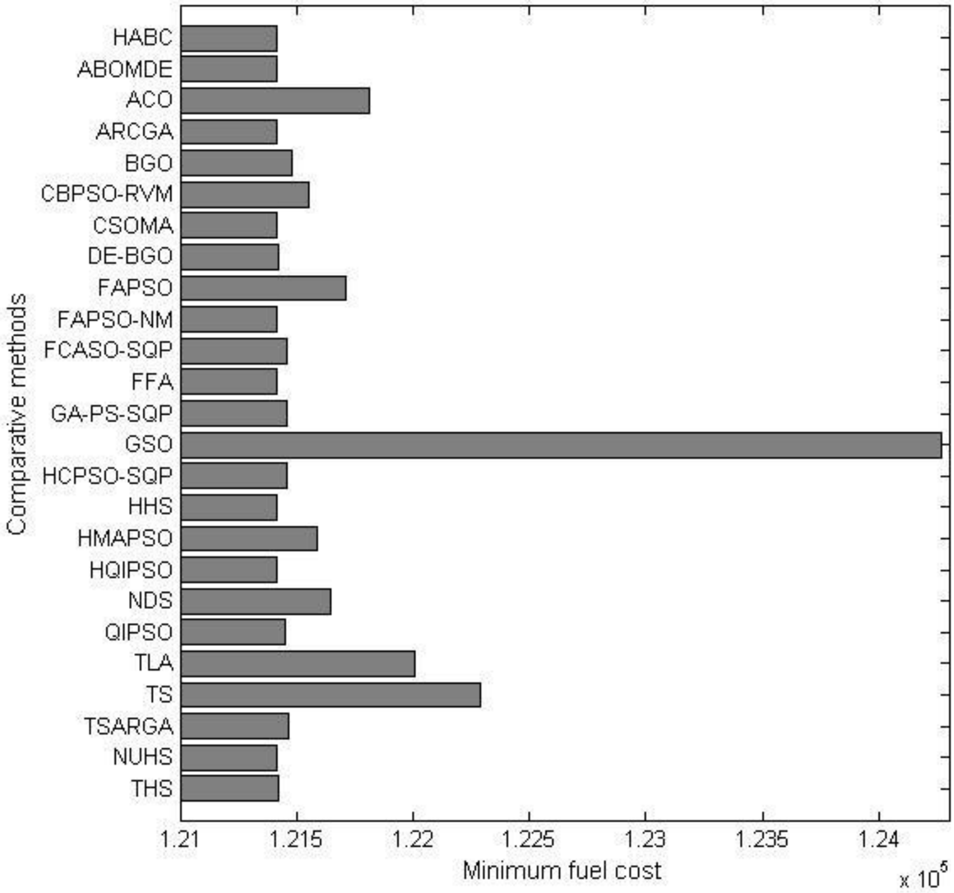

Figure 7: Minimum fuel cost comparison for Test System 3

Conclusion and future work

In this paper, the non-convex economic load dispatch (ELD) problem is solved using the hybridization of ABC algorithm with β-hill climbing optimizer, called HABC. The main objective of ELD problem is to minimize the total fuel cost production of the active generating units in the system. ELD is tackled by assigning generation levels to each generating unit in the system, subject to fulfillment the quality and inequality constraints. In HABC, the functionality of the onlooker bee phase is replaced with the β-hill climbing optimizer in order to empower its exploitation capability.

The proposed HABC is evaluated using three ELD problems with vary size and complexity. That are, two different ELD problems with thirteen generating units, and one problem with forty generating units. In order to analyze the sensitivity of the HABC, nine convergence scenarios are designed to reveal the effect of the parameter settings on the behavior of the proposed HABC.As shown from the experimental results, increasing the parameter value of *SN* and *limit*, and considerable value of βHCR leads to superior results. This validate in the experimental results reported in Tables 2, 3, 4, 6, 7, 8, 10, 11, and 12. Finally for comparative evaluation, the results obtained by the proposed HABC is compared with those achieved by the other methods from the literature. Interestingly, the proposed HABC achieved the second best results in the three cases of ELD problem.

In future, efforts will be made to re-evaluate HABC to other versions of ELD problems such as those with transmission losses and ramp rate limits. Furthermore, HABC can be further improved by improving the selection methods to be more focus on exploitation process. Other means of optimization problems with non-convex, non-linear and constrained nature can be also tackled using HABC to prove its performance.

References

AB WAHAB, M. N., NEFTI-MEZIANI, S. & ATYABI, A. 2015. A comprehensive review of swarm optimization algorithms. *PLoS One,* 10**,** e0122827.

AFZALAN, E. & JOORABIAN, M. 2015. An improved cuckoo search algorithm for power economic load dispatch. *International Transactions on Electrical Energy Systems,* 25**,** 958-975.

AKAY, B. & KARABOGA, D. 2015. A survey on the applications of artificial bee colony in signal, image, and video processing. *Signal, Image and Video Processing,* 9**,** 967-990.

AL-BETAR, M. A. 2017. $ $\beta $ $ β-Hill climbing: an exploratory local search. *Neural Computing and Applications,* 28**,** 153-168.

AL-BETAR, M. A., AWADALLAH, M. A., DOUSH, I. A., ALSUKHNI, E. & ALKHRAISAT, H. 2018. A Non-convex Economic Dispatch Problem with Valve Loading Effect Using a New Modified $ $\beta $ $-Hill Climbing Local Search Algorithm. *Arabian Journal for Science and Engineering***,** 1-18.

AL-BETAR, M. A., AWADALLAH, M. A., KHADER, A. T. & BOLAJI, A. L. A. 2016a. Tournament-based harmony search algorithm for non-convex economic load dispatch problem. *Applied Soft Computing,* 47**,** 449-459.

AL-BETAR, M. A., AWADALLAH, M. A., KHADER, A. T., BOLAJI, A. L. A. & ALMOMANI, A. 2016b. Economic load dispatch problems with valve-point loading using natural updated harmony search. *Neural Computing and Applications***,** 1-15.

ALSUMAIT, J., SYKULSKI, J. & AL-OTHMAN, A. 2010. A hybrid GA–PS–SQP method to solve power system valve-point economic dispatch problems. *Applied Energy,* 87**,** 1773-1781.

AWADALLAH, M. A., BOLAJI, A. L. A. & AL-BETAR, M. A. 2015. A hybrid artificial bee colony for a nurse rostering problem. *Applied Soft Computing,* 35**,** 726-739.

AZIZIPANAH-ABARGHOOEE, R., NIKNAM, T., ROOSTA, A., MALEKPOUR, A. R. & ZARE, M. 2012. Probabilistic multiobjective wind-thermal economic emission dispatch based on point estimated method. *Energy,* 37**,** 322-335.

BHATTACHARYA, A. & CHATTOPADHYAY, P. K. 2010a. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE transactions on power systems,* 25**,** 1955-1964.

BHATTACHARYA, A. & CHATTOPADHYAY, P. K. 2010b. Solving complex economic load dispatch problems using biogeography-based optimization. *Expert Systems with Applications,* 37**,** 3605-3615.

BLUM, C. & ROLI, A. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR),* 35**,** 268-308.

BOLAJI, A. L. A., KHADER, A. T., AL-BETAR, M. A. & AWADALLAH, M. A. 2013. Artificial bee colony algorithm, its variants and applications: A survey. *Journal of Theoretical & Applied Information Technology,* 47.

BOLAJI, A. L. A., KHADER, A. T., AL-BETAR, M. A. & AWADALLAH, M. A. 2014. University course timetabling using hybridized artificial bee colony with hill climbing optimizer. *Journal of Computational Science,* 5**,** 809-818.

BOUSSAID, I., LEPAGNOT, J. & SIARRY, P. 2013. A survey on optimization metaheuristics. *Information Sciences,* 237**,** 82-117.

CAI, J., LI, Q., LI, L., PENG, H. & YANG, Y. 2012a. A hybrid CPSO–SQP method for economic dispatch considering the valve-point effects. *Energy Conversion and Management,* 53**,** 175-181.

CAI, J., LI, Q., LI, L., PENG, H. & YANG, Y. 2012b. A hybrid FCASO-SQP method for solving the economic dispatch problems with valve-point effects. *Energy,* 38**,** 346-353.

CHAKRABORTY, S., SENJYU, T., YONA, A., SABER, A. & FUNABASHI, T. 2011. Solving economic load dispatch problem with valve-point effects using a hybrid quantum mechanics inspired particle swarm optimisation. *IET generation, transmission & distribution,* 5**,** 1042-1052.

COELHO, L. S. & MARIANI, V. C. 2006. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on power systems,* 21**,** 989-996.

DíAZ-CORTES, M.-A., CUEVAS, E. & ROJAS, R. 2017. Artificial Bee Colony Algorithm Applied to Multi-threshold Segmentation. *Engineering Applications of Soft Computing.* Springer.

DODU, J., MARTIN, P., MERLIN, A. & POUGET, J. 1972. An optimal formulation and solution of short-range operating problems for a power system with flow constraints. *Proceedings of the IEEE,* 60**,** 54-63.

DOS SANTOS COELHO, L. & MARIANI, V. C. 2009. An improved harmony search algorithm for power economic load dispatch. *Energy Conversion and Management,* 50**,** 2522-2526.

DOS SANTOS COELHO, L. & MARIANI, V. C. 2010. An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect. *Energy Conversion and Management,* 51**,** 2580-2587.

EL-KEIB, A., MA, H. & HART, J. 1994. Environmentally constrained economic dispatch using the Lagrangian relaxation method. *IEEE transactions on Power Systems,* 9**,** 1723-1729.

GAO, W., LIU, S. & HUANG, L. 2012. A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics,* 236**,** 2741-2753.

HE, Y., XIE, H., WONG, T.-L. & WANG, X. 2018. A novel binary artificial bee colony algorithm for the set-union knapsack problem. *Future Generation Computer Systems,* 78**,** 77-86.

HUSSAIN, K., SALLEH, M. N. M., CHENG, S. & SHI, Y. 2018. Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review***,** 1-43.

JABR, R. A., COONICK, A. H. & CORY, B. J. 2000. A homogeneous linear programming algorithm for the security constrained economic dispatch problem. *IEEE Transactions on power systems,* 15**,** 930-936.

KARABOGA, D. 2005. An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

KARABOGA, D., GORKEMLI, B., OZTURK, C. & KARABOGA, N. 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review,* 42**,** 21-57.

KHAMSAWANG, S. & JIRIWIBHAKORN, S. 2010. DSPSO–TSA for economic dispatch problem with nonsmooth and noncontinuous cost functions. *Energy Conversion and Management,* 51**,** 365-375.

KUMAR, R., SHARMA, D. & SADU, A. 2011. A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. *International journal of electrical power & energy systems,* 33**,** 115-123.

LIN, W.-M., CHENG, F.-S. & TSAY, M.-T. 2002. An improved tabu search for economic dispatch with multiple minima. *IEEE Transactions on power systems,* 17**,** 108-112.

LIN, W.-M., GOW, H.-J. & TSAI, M.-T. 2011. Combining of direct search and signal-to-noise ratio for economic dispatch optimization. *Energy conversion and management,* 52**,** 487-493.

LOHOKARE, M., PANIGRAHI, B. K., PATTNAIK, S. S., DEVI, S. & MOHAPATRA, A. 2012. Neighborhood search-driven accelerated biogeography-based optimization for optimal load dispatch. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* 42**,** 641-652.

LU, H., SRIYANYONG, P., SONG, Y. H. & DILLON, T. 2010. Experimental study of a new hybrid PSO with mutation for economic dispatch with non-smooth cost function. *International Journal of Electrical Power & Energy Systems,* 32**,** 921-935.

MANDAL, B., ROY, P. K. & MANDAL, S. 2014. Economic load dispatch using krill herd algorithm. *International journal of electrical power & energy systems,* 57**,** 1-10.

MEHMOOD, K. & AHMAD, A. 2017. Improved Grey Wolf Optimization for Economic Load Dispatch Problem Considering Valve Point Loading Effect and Prohibited Operating Zones. *The Nucleus,* 54**,** 250-257.

MENG, K., WANG, H. G., DONG, Z. & WONG, K. P. 2010. Quantum-inspired particle swarm optimization for valve-point economic load dispatch. *IEEE transactions on power systems,* 25**,** 215-222.

MORADI-DALVAND, M., MOHAMMADI-IVATLOO, B., NAJAFI, A. & RABIEE, A. 2012. Continuous quick group search optimizer for solving non-convex economic dispatch problems. *Electric Power Systems Research,* 93**,** 93-105.

NANDA, J., HARI, L. & KOTHARI, M. 1994. Economic emission load dispatch with line flow constraints using a classical technique. *IEE Proceedings-Generation, Transmission and Distribution,* 141**,** 1-10.

NETO, J. X. V., REYNOSO-MEZA, G., RUPPEL, T. H., MARIANI, V. C. & DOS SANTOS COELHO, L. 2017. Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution. *International Journal of Electrical Power & Energy Systems,* 84**,** 13-24.

NIKNAM, T., MOJARRAD, H. D., MEYMAND, H. Z. & FIROUZI, B. B. 2011. A new honey bee mating optimization algorithm for non-smooth economic dispatch. *Energy,* 36**,** 896-908.

PANDI, V. R., PANIGRAHI, B. K., MOHAPATRA, A. & MALLICK, M. K. 2011. Economic load dispatch solution by improved harmony search with wavelet mutation. *International Journal of Computational Science and Engineering,* 6**,** 122-131.

PANIGRAHI, B. & PANDI, V. R. 2008. Bacterial foraging optimisation: Nelder–Mead hybrid algorithm for economic load dispatch. *IET generation, transmission & distribution,* 2**,** 556-565.

POTHIYA, S., NGAMROO, I. & KONGPRAWECHNON, W. 2010. Ant colony optimisation for economic dispatch problem with non-smooth cost functions. *International Journal of Electrical Power & Energy Systems,* 32**,** 478-487.

SAYAH, S. & HAMOUDA, A. 2013. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. *Applied Soft Computing,* 13**,** 1608-1619.

SELVAKUMAR, A. I. & THANUSHKODI, K. 2007. A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE transactions on power systems,* 22**,** 42-51.

SHANG, Y., LU, S., GONG, J., LIU, R., LI, X. & FAN, Q. 2017. Improved genetic algorithm for economic load dispatch in hydropower plants and comprehensive performance comparison with dynamic programming method. *Journal of Hydrology,* 554**,** 306-316.

SINHA, N., CHAKRABARTI, R. & CHATTOPADHYAY, P. 2003. Evolutionary programming techniques for economic load dispatch. *IEEE Transactions on evolutionary computation,* 7**,** 83-94.

SUBBARAJ, P., RENGARAJ, R. & SALIVAHANAN, S. 2011. Enhancement of self-adaptive real-coded genetic algorithm using Taguchi method for economic dispatch problem. *Applied Soft Computing,* 11**,** 83-92.

SUNDAR, S., SUGANTHAN, P. N., JIN, C. T., XIANG, C. T. & SOON, C. C. 2017. A hybrid artificial bee colony algorithm for the job-shop scheduling problem with no-wait constraint. *Soft Computing,* 21**,** 1193-1202.

WALTERS, D. C. & SHEBLE, G. B. 1993. Genetic algorithm solution of economic dispatch with valve point loading. *IEEE transactions on Power Systems,* 8**,** 1325-1332.

YANG, X.-S., HOSSEINI, S. S. S. & GANDOMI, A. H. 2012. Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Applied soft computing,* 12**,** 1180-1186.

ZHANG, H., YUE, D., XIE, X., HU, S. & WENG, S. 2015. Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch. *Applied soft computing,* 34**,** 312-323.

ZHONG, F., LI, H. & ZHONG, S. 2017. An improved artificial bee colony algorithm with modified-neighborhood-based update operator and independent-inheriting-search strategy for global optimization. *Engineering Applications of Artificial Intelligence,* 58**,** 134-156.