

Article

Optimized Economic Load Dispatch with Multiple Fuels and Valve-Point Effects Using Hybrid Genetic–Artificial Fish Swarm Algorithm

Abdulrashid Muhammad Kabir ^{1,†}, Mohsin Kamal ^{2,†} , Fiaz Ahmad ^{3,†} , Zahid Ullah ^{4,*,†} ,
Fahad R. Albogamy ^{5,†} , Ghulam Hafeez ^{6,†}  and Faizan Mehmood ^{7,*,†} 

- ¹ Department of Electrical Engineering, Kebbi State University of Science and Technology, Aliero 863104, Nigeria; abdulrashidmkabir@yahoo.com
- ² KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia 2109, Cyprus; kamal.mohsin@ucy.ac.cy
- ³ Department of Electrical and Computer Engineering, Air University Islamabad, Islamabad 44000, Pakistan; fiaz.ahmad@mail.au.edu.pk
- ⁴ Department of Electrical Engineering, Sialkot Campus, University of Management and Technology (Lahore), Sialkot 51310, Pakistan
- ⁵ Computer Sciences Program, Turabah University College, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; f.ahmamdani@tu.edu.sa
- ⁶ Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad 44000, Pakistan; ghulamhafeez@uetmardan.edu.pk
- ⁷ Department of Electrical Engineering, University of Engineering and Technology, Taxila 47050, Pakistan
- * Correspondence: zahid.ullah@skt.umt.edu.pk (Z.U.); mehmood.faizan@ucy.ac.cy (F.M.)
- † These authors contributed equally to this work.



Citation: Kabir, A.M.; Kamal, M.; Ahmad, F.; Ullah, Z.; Albogamy, F.R.; Hafeez, G. and Mehmood, F. Optimized Economic Load Dispatch with Multiple Fuels and Valve-Point Effects Using Hybrid Genetic–Artificial Fish Swarm Algorithm. *Sustainability* **2021**, *13*, 10609. <https://doi.org/10.3390/su131910609>

Academic Editors: Sheraz Aslam, Herodotos Herodotou and Nouman Ashraf

Received: 9 August 2021
Accepted: 20 September 2021
Published: 24 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Abstract: Economic Load Dispatch (ELD) plays a pivotal role in sustainable operation planning in a smart power system by reducing the fuel cost and by fulfilling the load demand in an efficient manner. In this work, the ELD problem is solved by using hybridized robust techniques that combine the Genetic Algorithm and Artificial Fish Swarm Algorithm, termed the Hybrid Genetic–Artificial Fish Swarm Algorithm (HGAFSA). The objective of this paper is threefold. First, the multi-objective ELD problem incorporating the effects of multiple fuels and valve-point loading and involving higher-order cost functions is optimally solved by HGAFSA. Secondly, the efficacy of HGAFSA is demonstrated using five standard generating unit test systems (13, 40, 110, 140, and 160). Finally, an extra-large system is formed by combining the five test systems, which result in a 463 generating unit system. The performance of the developed HGAFSA-based ELD algorithm is then tested on the six systems including the 463-unit system. Annual savings in fuel costs of \$3.254 m, \$0.38235 m, \$2135.7, \$9.5563 m, and \$1.1588 m are achieved for the 13, 40, 110, 140, and 160 standard generating units, respectively, compared to costs mentioned in the available literature. The HGAFSA-based ELD optimization curves obtained during the optimization process are also presented.

Keywords: artificial fish swarm algorithm; economic load dispatch; genetic algorithm; hybrid genetic–artificial fish swarm algorithm; multi-objective optimization; sustainable power generating system



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern power systems around the world are becoming increasingly complex, with interconnections and varying load demands. There is an emergent need for power systems to be sustainable, reliable, low-cost, smarter, and cleaner, which would allow the broad participation of end users for energy generation and consumption and the management of loads by intelligent devices [1]. With this changing outlook, Economic Load Dispatch (ELD) is needed due to the lack of energy resources, increased power generation costs, and environmental concerns. In the actual scenario, the power plants are not equidistant from the load and there is no similar fuel cost function. Therefore, in order to provide cheaper

power, loads must be distributed to various power plants to minimize power generation costs. A practical economic dispatch (ED) problem has a highly nonlinear objective function with equality and inequality constraints. The ELD problem is solved using conventional methods such as lambda iteration, gradient methods, and non-conventional methods (heuristic methods). However, these technologies may not provide optimal solutions to find a global optimal solution as they require a piecewise linear and monotonically increasing incremental fuel cost curve. Optimal economic operation and planning of sustainable power generation systems is a very important pillar in the power industry [2]. ELD also refers to the operation of a power generation facility that produces energy at the lowest cost in order to recognize operational limitations of power generation facilities and provide reliable services to consumers. ELD schedules the output of available power generation units at specific times to minimize overall production costs while satisfying equality and inequality constraints [3]. Before 1973, due to the oil embargo measures that led to a sharp rise in fuel prices, utility companies spent around 20% of their total revenue on fuel for the generation of electric energy [4]. By 1980, this figure had increased to over 40% of the total revenue. In the five years following 1973, the fuel cost for electric utilities in the United States increased by 25% per year. Due to the fact that fuel is an irreplaceable natural resource, the efficient use of available fuels is of increasing importance [5].

Table 1 shows the parameters and assumptions for a moderately large power system [2]. The idea of size of money was obtained by considering the annual operation cost of a large utility for purchasing fuel. This cost reveals the direct requirement for customers' income to be an average of 3.15 cents/kWh, targeted for recovering fuel costs. Savings in the operation of a small part of the system refer not only to the amount of fuel consumed but also to the substantial reduction in operating costs. Therefore, this field has gained tremendous attention from engineers for many years. However, a regular change in the basic fuel price level plays the role of emphasizing the problem and enhancing its economic significance. Inflation also poses problems in developing economic operational techniques, methods, and examples of power generation systems [6]. Moreover, the rapid increase in the size and power demand of power systems resulted in reduced operating costs while maintaining the thermal limitations of voltage security and transmission line branching. Many mathematical programming and artificial intelligence techniques, such as GA-based ELD, Particle Swarm Optimization (PSO)-based ELD [7,8], ELD based on dynamic programming and evolutionary programming, and hybrid GA-PSO-based ELD, are applied to solve the aforementioned problem. In the most common formulation, the ELD problem is modeled as a large-scale, non-convex, nonlinear, static optimization problem in both discrete and continuous control variables [9].

Table 1. Parameters and assumptions for total annual fuel cost.

Parameters		Assumptions	
Annual peak load and load factor	10,000 MW and 60 %	Annual energy produced	$10^7 \text{ MW} \times 8760 \text{ h/year} \times 0.60$ $= 5.256 \times 10^{10} \text{ kWh}$
Average annual heat rate for converting fuel to electric energy	10,550.56 KJ/kWh	Annual fuel consumption	$10,550.56 \text{ KJ/kWh} \times 5.256 \times 10^{10} \text{ kWh} = 55.45 \times 10^{13} \text{ KJ}$
Annual fuel cost (corresponds to oil price at \$18/bbl)	\$3.00/1.055 GJ	Annual fuel cost	$55.45 \times 10^{13} \times 3/1.055 \times 10^{-9} \text{ \$/J} = \$1.5767 \text{ million}$

Many researchers [6–9] have modeled the nonlinear, convex nature of ELD problems using pure quadratic functions, with the quadratic coefficients (a, b, and c) defined at the start of the solution search process. Meanwhile, research work [10] developed realistic models to incorporate the effect of multiple fuel cost functions and valve-point loading into the formulation of the ELD problem. For example, the authors in [8] employed PSO with the BAT algorithm for solving ELD problems considering inspired acceleration coefficients. In [9], the authors analyzed the generating unit profiles of various distributed generation

systems of different technologies. The energy loss was minimized for a distribution system with a mix of renewable energy resources using optimization techniques [10]. The authors in [11] proposed dynamic ELD in a sustainable power system using an accurate forecasting model and improved salp-swarm optimizer considering PV, energy storage, the power system, and various constraints of generating units. In [12], the authors proposed a combined dispatch strategy (load following and cycle charging) for the energy management and optimal operation of a hybrid energy system. A gradient-based optimizer inspired by the Newton technique was used for solving the ELD problem considering the valve-point effect [13]. The authors in [14] proposed a teaching-learning optimization for dynamic ELD of wind energy and load demand uncertainties considering various operation constraints. Most of the research works utilized various optimization techniques to minimize the fuel costs, reduce environmental concerns, and ensure dynamic ELD. In this regard, this work proposes a hybrid optimization technique to achieve the aforementioned objectives.

The key contribution of this work is to address the ELD problem by developing a more realistic model and by considering the effects of valve-point loading and multiple fuel cost functions. A multi-objective optimization problem is formulated that minimizes the fuel cost of generating units and the amount of nitrogen-bearing (NO_x) gases emitted by the generating units during their operation. Therefore, there is an emergent need to employ robust techniques to provide a reliable solution to the aforementioned complex optimization problem. Some have used heuristic techniques in an attempt to solve the abovementioned optimization technique to some extent, but this did not provide guaranteed efficient solutions. In order to provide a reliable and efficient solution, a hybridization of two conventional heuristic techniques (GA and AFSA), called HGAFSA, is proposed in order to solve the complex multi-objective optimization problem. HGAFSA is applied to solve a multi-objective ELD problem considering the effects of multiple fuel cost functions and valve-point loading. The effectiveness of the proposed approach is demonstrated using five standard generating unit test systems (13, 40, 110, 140, and 160) and a 463 generating unit system formed by the combination of the five systems, and the results are compared with the best results presented in the literature. The choice of GA and AFSA is based on the following factors: (a) GA is a heuristic technique with a well-defined set of search equations that is effective in solving problems such optimal sizing and location of capacitor banks and distributed generators [15], optimal power flow [16], optimal location of tie and sectionalizing switches in distribution systems, and optimal network expansion [17], and (b) AFSA is a relatively new heuristic technique based on well-refined and sophisticated solution search equations and is widely applied in controller design, optimal PID tuning, and objective function minimization/maximization [18].

The major contributions of this work to the existing body of knowledge are summarized as follows:

1. HGAFSA, capable of solving a higher-order ELD, is developed and used to solve several higher-order ELD problems, including 13, 40, 110, 140, 160, and 463 unit systems.
2. An ELD encoder algorithm is developed and linked to the developed HGAFSA to form a HGAFSA-based ELD algorithm that minimizes any ELD cost function better than every algorithm mentioned in the available literature.
3. The effectiveness of the developed HGAFSA is demonstrated on six ELD systems, including the 463-unit system. Annual savings in fuel costs of \$3.254 m, \$0.38235 m, \$2135.7, \$9.5563 m, and \$1.1588 m for the 13, 40, 110, 140, and 160 units, respectively, for the first five systems are achieved, compared to costs reported in the available literature.

The remainder of the paper is organized as follows. Section 2 formulates the ELD problem. The formulation of the proposed HGAFSA is presented in Section 3. Section 4 provides the performance validation and simulation settings of the proposed system. The paper concludes with a brief summary and future directions in Section 5.

2. Formulation of ELD

The purpose of the ELD problem is to find an optimal combination of power generation that meets the constraints of equality and inequality while minimizing the total power generation cost. The fuel cost curve for any unit is an approximation of the quadratic function segment of the generator's active power output by assumption [19].

2.1. The Cost Function

Cost function is a financial term used by a company's economists and managers as a way to express how different costs differ under different circumstances. This shows how to display monetary output. Changes in the level of activities related to these outputs will change rates and fees from overhead and operating expenses [19]. The linear cost function has three basic types:

1. Functions of fixed cost (defined by a straight line with a zero (0) gradient).
2. Functions of variable cost (defined by a straight line with positive gradient and having no intercept).
3. Functions of mixed cost (defined by a line having single or multiple gradient(s) and intercept(s)).

In mixed environments, costs are fixed to specific points, which can be changed based on related activities. Analysts use this type of function to make important predictions about the market and to inform various decision-making tasks [19]. For a given network of power generating units, a set of cost functions are usually defined based on the mixed function types to account for the operational cost of each generating unit. Conventionally, the quadratic form of the cost functions is most widely used [20]. A quadratic cost function is a mixed cost function that partly comprises a single fixed cost function (represented by a fixed coefficient, e.g., a) and two variable cost functions (represented by products of coefficients b and c , and functions of output power P and P^2). For a network of n generating units, the overall cost function (F_T) of the system is a summation of the n individual cost functions [$F(P_1), F(P_2), \dots, F(P_{n-1}), F(P_n)$] of the various units in the system. This can be simply represented by Equation (1). An ELD problem is an optimization problem that is aimed at minimizing the F_T subject to a set of operating constraints [21].

$$F_T = \sum_{i=1}^n F(P_i) = \sum_{i=1}^n a_i + b_i P_i + c_i P_i^2. \quad (1)$$

In general, F_T is used to represent the fuel or emission cost (\$/hr) depending on the values of the coefficients (a_i , b_i and c_i) used [21]. In order to ensure proper operation of the network of generating units, it is essential to ensure that a set of useful constraints are satisfied. These constraints include the generator capacity (inequalities) and active power balance (equality) constraints given in Equations (2) and (3), respectively [21].

$$P_{i,min} \leq P_i \leq P_{i,max} \quad \text{for } i = 1, 2, \dots, n, \quad (2)$$

where $P_{i,min}$ and $P_{i,max}$ are the minimum and maximum power output of the i th unit.

$$P_D = \sum_{i=1}^n P_i - P_{Loss}. \quad (3)$$

where P_D is the total power demand and P_{Loss} is the total transmission loss. The transmission loss P_{Loss} can therefore be calculated by using the B matrix technique and is defined by Equation (4) as follows [21]:

$$P_{Loss} = \sum_{i=1}^n \sum_{j=1}^n P_i B_{ij} P_j, \quad (4)$$

where B_{ij} s are the elements of loss coefficient matrix B. The cost function defined by F_T in Equation (1) assumes a smooth quadratic fuel cost function without the valve-point loadings of the generating units (where the valve-point effects are ignored). The generating units with multi-valve steam turbines exhibit greater variation in the fuel cost functions. Since the valve point results in ripples, a cost function with higher-order nonlinearity will result. Therefore, the function $F(P_i)$ in Equation (1) can be replaced by Equation (5) to account for the valve-point effects [22]. Conventionally, sinusoidal functions are often added to the quadratic cost function to account for the valve-point effect, as given in Equation (5).

$$F(P_i) = a_i + b_i P_i + c_i P_i^2 + |e_i \times \sin(f_i \times (P_{i,min} - P_i))| . \quad (5)$$

where e_i and f_i are the cost coefficients of the i th unit with valve-point effects. In general, the cost coefficients e_i and f_i are introduced as in Equation (5) to model the valve-point loadings. Similarly, Equation (5) can be used to represent either the fuel or emission cost (\$/hr) depending on the values of the coefficients (a_i , b_i , c_i , e_i , and f_i) that are used. Finally, Equation (5) also represents the proposed higher-order cost function considering the valve-point loading effect. In practical situations, generating units are made up of subunits. These subunits combine to give rise to the overall installed capacity of the unit. Most units are designed to operate using more than one fuel type, particularly in the case in which there is great fluctuation in the price and availability of the dominant fuel types [19]. In the case of moderately large units, a combination of the available fuel types may be used to cover the power demand over the specified period of time. This type of scenario introduces greater nonlinearity into the overall fuel cost function. Therefore, the fuel cost function of such a system can be modeled using a multiple fuel cost function, which is only defined for a particular range of power output within the specified maximum and minimum power generation. Considering both the valve-point loading effect and multiple fuels, the cost function of the system may easily be represented using Equation (6) [19].

$$F(P_i) = \begin{aligned} & a_{i1} + b_{i1} P_i^2 + c_{i1} P_i^2 + |e_{i1} \times \sin(f_{i1} \times (P_{i1,min} - P_{i1}))|_{Fuel1 : P_i^{min} \leq P_i \leq P_{i1}} \\ & a_{i2} + b_{i2} P_i^2 + c_{i2} P_i^2 + |e_{i2} \times \sin(f_{i2} \times (P_{i2,min} - P_{i2}))|_{Fuel2 : P_i^{min} \leq P_i \leq P_{i2}} \\ & a_{ik} + b_{ik} P_i^2 + c_{ik} P_i^2 + |e_{ik} \times \sin(f_{ik} \times (P_{ik,min} - P_{ik}))|_{Fuel3 : P_{ik-1}^{min} \leq P_i \leq P_{ik}^{max}} . \end{aligned} \quad (6)$$

2.2. Artificial Fish Swarm Algorithm

AFSA guarantees a global optimum in solution search problems [23], which is of great importance in artificial intelligence to perform behavioral modeling. Consider a swarm consisting of N artificial fishes and a state vector $X = (x_1, x_2, \dots, x_n)$, where n states or attributes of the artificial fish are to be optimized via the AFSA algorithm. In addition, suppose that $Y = f(X)$ represents the objective function giving the food concentration of the artificial fish at the current position, and let $D_{ij} = ||X_i - X_j||$ be used to describe the distance between artificial fishes i and j . Other important parameters for the artificial fish, including its vision field, maximum step for motion, the congestion factor, and maximum attempts in each praying, are also taken into account and are expressed as visual, step, δ , and try number, respectively. For better results, the congestion factor is used to constrain the size of the artificial swarm [5]. The behavior of the artificial fish is described next as praying, swarm, and chasing.

2.2.1. Praying

If the artificial fish is currently in state X_i , in order to carry out praying, then it must select another state, e.g., X_j , that is located within its visual field. Afterwards, the search for a minimal solution is continued until $Y_i \geq Y_j$; if this is the case, praying is completed by moving one step in the direction taken. However, if $Y_i \leq Y_j$, another state X_j must be reselected from the visual field randomly to analyze whether it can move forward based on a certain forwarding condition. This procedure is repeated for try-number times, and if the

forward motion condition is still not satisfied, it will take one step in a random direction. Mathematically, this can be expressed as given in Equation (7);

$$\begin{aligned}x_{i-next-k} &= x_{i \rightarrow k} + \frac{x_{jk} - x_{ik}}{\|X_j - X_i\|} * random(step) \quad Y_j > Y_i, \\x_{i-next-k} &= x_{i \rightarrow k} + random(step) \quad Y_j \leq Y_i,\end{aligned}\quad (7)$$

where $k = 1, 2, \dots, n$, x_{ij} represents the k th element of X_i , which is the current state of the artificial fish; x_{jk} is the k th element of X_j , which is the state of the artificial fish after random movement, and $x_{i-next-k}$ represents the k th element of x_{i-next} , which is the next state of the artificial fish. Similarly, Y_i and Y_j are the values of the objective function of the current state and that after a random movement, respectively, and $random(step)$ represents a random number selected from the range defined by [0 step].

2.2.2. Swarm

In the swarming process, the fish has the natural ability to share food and avoid any distraction that is encountered. Suppose that the current state of the artificial fish is given by X_j , and the total number of other fishes in its vision domain is denoted by n . Now, if $n_f = 0$, this should mean that the visual domain of the given artificial fish is empty, so it is time to implement praying. However, if $n_f \geq 0$, this means that there are other companion fishes present in its vision domain, and it must start searching the central position X_c (i.e., center between the present fishes) of its companions according to Equation (8) [24].

$$X_{ck} = \frac{(\sum_{j=1}^{n_f} x_{jk})}{n_f}, \quad (8)$$

where X_c represents the central position of the artificial fish among other fishes, X_{ck} gives the k th element of X_c , and X_{jk} denotes the k th element of the vector of j th companion $j = (1, 2, \dots, n)$. The calculation of the food concentration of the artificial fish at the central position, given by Y_c , is the objective function with the constraint of $Y_c \frac{n_f}{Y_i} > 1$. If the central position is less congested and safer, the artificial fish must move towards this position using Equation (9); otherwise, praying is implemented [24].

$$x_{i-next-k} = x_{i \rightarrow k} + \frac{x_{ck} - x_{ik}}{\|X_c - X_i\|} * random(step). \quad (9)$$

2.2.3. Chasing

In an artificial fish swarm, when fishes are in search of food, neighboring partners have the natural ability to trace and reach food more quickly. Suppose that X_i denotes the current state of the artificial fish and n denotes the total number of companions in its visual field. Now, if $n_f = 0$, this shows that the visual field of the artificial fish is empty; therefore, praying should be implemented. However, if $n_f \geq 1$, this indicates that some companions do exist in its visual field; therefore, it should search and find a companion with a minimum value of the corresponding function X_{max} . Then, the constraint is checked, i.e., $Y_{max} \frac{n_f}{Y_i} > 1$; if it is valid, this means that the fitness value of the corresponding companion is small and it is not congested; thus, Equation (10) is implemented; otherwise, praying is implemented [24].

$$x_{i-next-k} = x_{i \rightarrow k} + \frac{x_{max,k} - x_{ik}}{\|X_{max} - X_i\|} * random(step). \quad (10)$$

where $x_{max,k}$ gives the k th element of state vector X_{max} .

2.3. Genetic Algorithm

Genetic algorithm (GA) is an optimization algorithm that simultaneously works on several solutions (also called population), as opposed to other optimization methods that work on one solution at a time [25,26]. It is an iterative optimization algorithm and comprises several steps, briefly described below.

2.3.1. Reproduction

The foremost operation on a population is called reproduction, which establishes a mating pool by the selection of good strings from a population. The mating pool is fed with duplicate copies of the good strings, i.e., above average strings. Proportionate selection of strings from the present population is the most common operation in the reproduction process, where each string is selected based on its fitness probability. Hence, an i th string is selected with a fitness probability of ϵ_i . The cumulative probability of all the strings in a population is always '1' because the population size in GA is normally fixed. The fitness probability of i th string is given as $\frac{f_i}{\sum_{j=1}^N f_j}$. N represents the size of the population. In [27], the authors have presented a method for achieving a proportionate selection using a roulette wheel, where the circumference for every string is marked exactly according to its fitness.

2.3.2. Crossover

Crossover is the next operation applied to the string of the mating pool after the reproduction operation. In this operation, two strings are selected randomly from the mating pool and some of their portions are exchanged. For instance, in a single point crossover operation, two new strings are produced by swapping the right-side portions of two strings after cutting these at arbitrary places, as presented in [15].

A better child string can be produced by combining good sub-strings from either parent if a suitable site is selected, which is usually selected randomly since the suitability of a site is not always known [15]. However, it must be noted that the random selection of a site does not make the search process random. If a single-point crossover is applied to two 1-bit strings from either parent, at most, different strings can be found as a solution in the given search space. With the selection of a random site, 2^i children strings are produced. These strings may or may not contain good sub-strings from parent strings, which depends on the selection of an appropriate site. This aspect is of little interest as, if the crossover operation fails to produce good strings, the reproduction operation will produce more copies in the following mating pool. In a similar way, with a two-point crossover operation, two sites are chosen randomly. A multi-point crossover operation can be carried out in a similar fashion, and this extension is usually called the uniform crossover operator.

For a case of binary strings, the uniform crossover operation is applied by selecting from either parent every bit with a probability of 0.5 [15]. The major purpose of the crossover operation is the search of the parameter space and the preservation of the information from parent strings since these are labeled as good strings after due selection by the reproduction operation. Maximum information is transferred or preserved from parent to child strings with a single-point crossover operator search as opposed to that of the uniform crossover operator, where the search is extensive but the information preservation from parent to child is minimal. For a crossover probability of P_c , the crossover operation is applied to $100P_c\%$ of the strings in the population and the remaining strings, i.e., $100(1-P_c)\%$ are transferred to a new population [15].

2.3.3. Mutation

In genetic algorithms, search procedures are normally carried out with the crossover operator. However, a mutation operation may also be used sometimes for this purpose, which uses certain mutation probability P_m , to change a 1 to 0 and vice versa. The detailed process of mutation is explained in [15] where a new solution is created after a change in the value of the fourth gene. The mutation operation is necessary to diversify the

population, which can be demonstrated with the above example, where a long string of zeros occurs, and a 1 is required to obtain a new solution that is more optimal or near to optimal. Mutation is also helpful in improving a local solution.

3. Formulation of Hybrid Genetic–Artificial Fish Swarm Algorithm

The proposed HGAFSA is designed based on the available parameter dredging steps present in the conventional GA and AFSA. However, each of the separate algorithms (GA and AFSA) is assumed to be composed of three major steps, as described below.

1. GA
 - (a) Reproduction;
 - (b) Crossover;
 - (c) Mutation.
2. AFSA
 - (a) Praying;
 - (b) Swarming;
 - (c) Chasing.

The mathematical formulation of these steps has been described in Section 2. The proposed HGAFSA is a logical combination of the six steps listed above. It is worth noting that the GA uses a binary operation on a set of binary codes known as chromosomes, which further comprise genes, whereas the AFSA uses real numbers ranging between zero and one as the parameters of the so-called artificial fish. As such, a decoder function and an encoder function are required to serve as converters from GA to AFSA and vice versa. These functions are intended to decode binary code into real numbers and later encode real numbers into binary.

3.1. Decoder Function

The decoder function takes in four parameters as inputs and generates a decoded version of the main parameter as the output. Here, the main parameter is the chromosome (X), whereas the remaining three parameters are:

1. x_{min} → Lower boundary of the desired decoded output;
2. x_{max} → Higher boundary of the desired decoded output;
3. N_{bits} → Number of bits per parameter.

The steps involved in decoding a single chromosome can be described using Algorithm 1.

Algorithm 1: The decoder function

input : X, x_{min}, x_{max} and N_{bits}
output: $X_{decoded}$

- 1 **Evaluate**: $N = \text{number of bits in } X$;
- 2 **Evaluate**: $p = N * (N_{bits})^{-1}$ //number of parameters;
- 3 **Evaluate**: $q = 0.5^{[1,2,\dots,N_{bits}]}$ //quantization levels;
- 4 **Evaluate**: $q_{norm} = q * (\sum_{i=1}^{N_{bits}} q_i)^{-1}$ //quantization level normalization;
- 5 **Evaluate**: $X_{decoded}(i) = [q(1) * X(j+1)q(2) * X(j+2)\dots q(N_{bits} * X(j + N_{bits}))]$;
- 6 **for** $i \leftarrow 1$ **to** p **and** $j \leftarrow (i - 1) * N_{bits}$ **do**
- 7 $X_{decoded} = [X_{decoded}(1)X_{decoded}(2)\dots X_{decoded}(p)] * (x_{max} - x_{min}) + x_{min}$;

Using the decoder function, a vector X with $p \times N_{bits}$ elements is decoded into a vector $X_{decoded}$ with p elements. As an illustration, consider $X = [1101101101110111011110111]$, if $x_{min} = 0$ and $x_{max} = 1$, let $N_{bits} = 6$. Algorithm 1 yields $[0.8571 \ 0.8730 \ 0.4762 \ 0.9365 \ 0.9365]$. In the proposed HGAFSA, once any of the GA steps is executed, the resulting output/population must be decoded before their respective fitnesses can be evaluated. Meanwhile, the resulting output/population from the AFSA steps are directly evaluated using the fitness function without being decoded.

3.2. Encoder Function

The encoder function is also written to counteract the effect of the decoder function presented in Algorithm 1. However, a reversed procedure is adopted based on Algorithm 1 (moving from step 7 to 1). Here, the decoder function is intended to generate chromosome (X) given its decoded version $X_{decoded}$, x_{min} , x_{max} and N_{bits} . However, the encoder function formulation is omitted from this manuscript for brevity. Generally, it can be said that the decoder function converts a chromosome into a fish, whereas the encoder function converts a fish back into a chromosome. This can be further described using Figure 1. Furthermore, it is worth noting that both X and $X_{decoded}$ are kept for reference during the optimization process using the proposed HGAFSA. However, either X or $X_{decoded}$ is later discarded depending on which of the GA or AFSA steps perform better at a given generation and at a given step in the HGAFSA dredging process. At first, the entire population is stored as chromosomes. However, each chromosome is either left as a X or transformed into a fish $X_{decoded}$ depending on which of the HGAFSA steps (GA step or AFSA step) performs better.

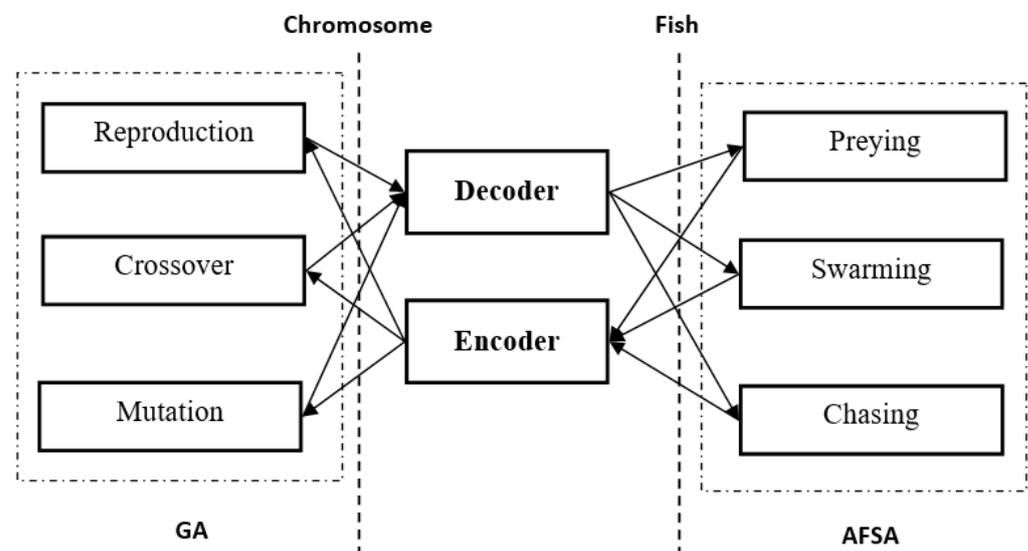


Figure 1. Illustration of chromosome to fish conversion and vice versa.

3.3. Population Update

The proposed HGAFSA is composed of two unique algorithms (GA and AFSA) with completely different parameter dredging procedures. When one of the GA steps is executed on an encoded fish (chromosome), the resulting chromosome might be of poor fitness compared to the result if an AFSA step had been directly performed on the fish itself. However, this consequence might be reversed. Therefore, the population must be carefully updated for optimality and an improved convergence rate. Algorithm 2 further describes the population update procedure.

Algorithm 2: Population updated

```

input : //list of steps to be executed in chronological order)
        combined population of fish and chromosomes
output:  $P(k)$ 
1 Define:  $\Omega =$ ;
2 for  $k \leftarrow 1$  to  $popsiz$  do
3   for  $s \leftarrow 1$  to  $N_{steps}$  do
4     if  $P(k) = fish \ \& \ S(s) \ AFSA$  then
5       Evaluate:  $f = F_{obj}(P(k))$ ;
6       Execute:  $P(k) \rightarrow S(s) \rightarrow P(k)_{new}$ ;
7       Evaluate:  $f_{new} = F_{obj}(P(k)_{new})$ ;
8       if  $f_{new}$  is better than  $f$  then
9         Store:  $[f_{new}P(k)_{new}] \rightarrow \Omega$ ;
10      else
11        Store:  $[fP(k)] \rightarrow \Omega$ ;
12      else if  $P(k) = fish \ \& \ S(s) \ GA$  then
13        Evaluate:  $f = F_{obj}(P(k))$ ;
14        Execute:  $P(k) \rightarrow Encoder \rightarrow P(k)_{Encoded}$ ;
15        Execute:  $P(k)_{Encoded} \rightarrow S(s) \rightarrow P(k)_{Encoded,new}$ ;
16        Evaluate:  $f_{new} = F_{obj}(P(k)_{Encoded,new})$ ;
17        if  $f_{new}$  is better than  $f$  then
18          Store:  $[f_{new}P(k)_{Encoded,new}] \rightarrow \Omega$ ;
19        else
20          Store:  $[fP(k)] \rightarrow \Omega$ ;
21      else if  $P(k) = chromosome \ \& \ S(s) \ AFSA$  then
22        Execute:  $P(k) \rightarrow Decoder \rightarrow P(k)_{Decoded}$ ;
23        Evaluate:  $f = F_{obj}(P(k)_{Decoded})$ ;
24        Execute:  $P(k)_{Decoded} \rightarrow S(s) \rightarrow P(k)_{Decoded,new}$ ;
25        Evaluate:  $f_{new} = F_{obj}(P(k)_{Decoded,new})$ ;
26        if  $f_{new}$  is better than  $f$  then
27          Store:  $[f_{new}P(k)_{Decoded,new}] \rightarrow \Omega$ ;
28        else
29          Store:  $[fP(k)] \rightarrow \Omega$ ;
30      else if  $P(k) = chromosome \ \& \ S(s) \ GA$  then
31        Execute:  $P(k) \rightarrow Decoder \rightarrow P(k)_{Decoded}$ ;
32        Evaluate:  $f = F_{obj}(P(k)_{Decoded})$ ;
33        Execute:  $P(k) \rightarrow S(s) \rightarrow P(k)_{new}$ ;
34        Execute:  $P(k) \rightarrow Decoded \rightarrow P(k)_{new,Decoded}$ ;
35        Evaluate:  $f_{new} = F_{obj}(P(k)_{new,Decoded})$ ;
36        if  $f_{new}$  is better than  $f$  then
37          Store:  $[f_{new}P(k)_{new,Decoded}] \rightarrow \Omega$ ;
38        else
39          Store:  $[fP(k)] \rightarrow \Omega$ ;
40 Sorting: Rearrange the vectors in  $\Omega$  (in the order of optimality);

```

3.4. Model of the Economic Load Dispatch Problem

To solve the ELD problem using the proposed HGAFSA, a function is required to convert the random numbers generated by it into electrical power demand scheduled to the set generating units. Let P be a set of power to be generated by the available generating units forming the ELD problem. Let P_{max} and P_{min} be the maximum and minimum power allowable for each of the units, respectively. N_G is the number of generating units. Then,

$$P = [P_1 P_2 P_3 \dots P_{N_G-2} P_{N_G-1} P_{N_G}], \quad (11)$$

$$P_{max} = [P_{max,1} P_{max,2} \dots P_{max,N_G-1} P_{max,N_G}], \quad (12)$$

$$P_{min} = [P_{min,1} P_{min,2} \dots P_{min,N_G-1} P_{min,N_G}]. \quad (13)$$

The next most important parameter of ELD problem formulation is the total power (P_T) to be generated by the generating units to meet both the power demand and the

power losses along the network. This can be described using the equality constraint as in Equation (14).

$$P_T = P_D + P_L . \tag{14}$$

where P_D is the total power demanded by the consumers and P_L is the total power losses in the network. P_T can also be expressed using Equation (15).

$$P_T = \sum_i^{N_G} P_i . \tag{15}$$

As described earlier, to evaluate the fitness of a population generated by HGAFSA, the population must be decoded into a fish ($X_{decoded}$). However, this fish must be further converted into a real power demand P . To achieve this, let $X_{decoded}$ be replaced by χ having p elements, such that Equation (16) holds.

$$\chi = x_1 x_2 x_3 \dots x_{p-2} x_{p-1} x_p . \tag{16}$$

An ELD encoder function is developed to transform χ into an equivalent P . The overall process can be described using Figure 2.

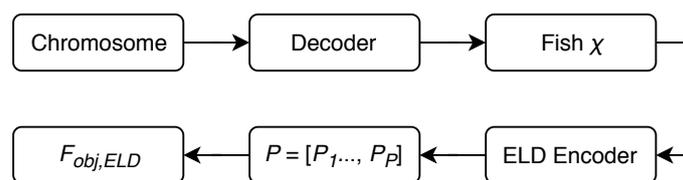


Figure 2. Evaluating the fitness of a chromosome/fish.

An ELD encoder function is developed to transform χ into an equivalent P . The overall process can be described using Figure 2. The function $F_{obj,ELD}$ is the objective function of the ELD problem. The ELD encoder function is given in Algorithm 3. The proposed ELD encoder has the advantage that no generating unit can generate below its minimum allowable generating limit. However, its generation may exceed the allowable maximum. To prevent this, steps 11 to 14 are added to Algorithm 3 to enforce the maximum limit constraint. Furthermore, in the developed ELD encoder, the power generation at any unit cannot be lower than the minimum allowable generation at that unit. This is because, in line 10 of the ELD encoder Algorithm, the addition of $P_{min}(h)$ enforces the lower limit constraint. However, the parameter γ can be greater than the upper limit at a particular generating unit due the presence of a rational function at the first part. The rational part of γ usually results in approximation, pushing its value beyond the maximum allowable limit. This phenomenon can be avoided by replacing γ with $P_{max}(h)$ when the upper limit is exceeded or using line 11 to 14 of Algorithm 3.

However, it can be observed from step 16 of Algorithm 3 that the proposed ELD encoder performs optimization during the encoding process. Therefore, it could be termed an optimal ELD encoder of order p . To further increase the performance of the proposed ELD encoder, the dimension of X can be extended for every given χ generated by the proposed HGAFSA. This extension is modeled in Equation (17).

$$\chi_{ext} = [\chi, \chi |(\sin(2\pi\chi))|, \chi |(\cos(2\pi\chi))|] . \tag{17}$$

With this extension, a new optimal ELD encoder of order $3 \times p$ can be formed by modifying steps 6 and 14 of the former optimal ELD encoder (order p) described earlier. This can be achieved using Equations (18) and (19), respectively.

$$par = [1, 2, 3, \dots, 3 \times p] , \tag{18}$$

$$s = [1, 2, 3, \dots, 3 \times p]. \quad (19)$$

Equation (17) can be used to boost the performance of HGAFSA by further exploring the solution search space. This is because, for every new set of possible solutions χ , two additional solution sets are also searched, i.e., $\chi_1 = \chi |(\sin(2\pi\chi))|$ and $\chi_2 = \chi |(\cos(2\pi\chi))|$. Therefore, the probability that the optimal solution will lie within the three search spaces (χ , χ_1 , and χ_2) must always be greater than or equal to the probability of the obtained optimal solution being within χ alone. As such, the performance of the original ELD encoder (which makes use of χ alone) will definitely improve. In the other hand, the ELD encoder of order $3 \times p$ is of a higher rate of convergence than the original ELD encoder (with order p).

Algorithm 3: ELD Encoder

```

input :  $\chi, p, N_G, P_{max}, P_{min}$  and  $P_T$ 
output:  $P$ 
1 Create:  $\zeta = []$ ;
2 for  $g = [1, 2, 3, \dots, N_G]$  do
3    $\partial = (P_{max}(g) - P_{min}(g)) * \chi / \max(\chi)$ ;
4    $\zeta = \begin{bmatrix} \xi \\ \partial \end{bmatrix}$ ;
5 Create:  $\varepsilon = []$ ;
6 for  $par = [1, 2, 3, \dots, p]$  do
7   for  $g = [1, 2, 3, \dots, N_G]$  do
8      $\rho(g) = \zeta(g, par)$ ;
9   for  $h = [1, 2, 3, \dots, N_G]$  do
10     $\gamma = \frac{(P_T - \sum_g^{N_G} P_{min}(g)) * \rho}{(\sum_g^{N_G} \rho(g))} + P_{min}(h)$ ;
11    while  $\max(\gamma - P_{max}) > 0$  (enforce the upper limit constraint for each generating unit) do
12       $\rho = \text{rand}(1, N_G) * \rho + \text{rand}(1, N_G)$ ;
13      repeat
14        | for loop at step 9;
15      until;
16     $\varepsilon = \begin{bmatrix} \xi \\ \gamma \end{bmatrix}$ ;
17 for  $s = [1, 2, 3, \dots, p]$  do
18   Evaluate:  $\text{Fit}(s) = F_{obj,ELD}([\varepsilon(s,1) \ \varepsilon(s,2) \ \varepsilon(s,3) \ \dots \ \varepsilon(s, N_G - 1) \ \varepsilon(s, N_G)])$ ;
19 Determine:  $K \leftarrow \text{Fit}(K) = \min(\text{Fit})$ ;
20 Assign:  $P = [\varepsilon(K,1) \ \varepsilon(K,2) \ \varepsilon(K,3) \ \dots \ \varepsilon(K, N_G - 1) \ \varepsilon(K, N_G)]$ ;

```

3.5. The Proposed HGAFSA-Based Higher-Order ELD Algorithm

The ELD problem is one of the power system analysis problems with a large number of possible solutions. However, such solutions form a set of local optimums. As the number of generating units increases, the possible solutions increase exponentially. As such, an algorithm that can deeply search into the solution domain is required to locate the global optimum solution. In this work, the HGAFSA optimization algorithm with high computation capability and a fast rate of convergence is developed for complex ELD problem solving. The flow chart for the proposed HGAFSA-based higher-order ELD problem solver is shown in Figure 3.

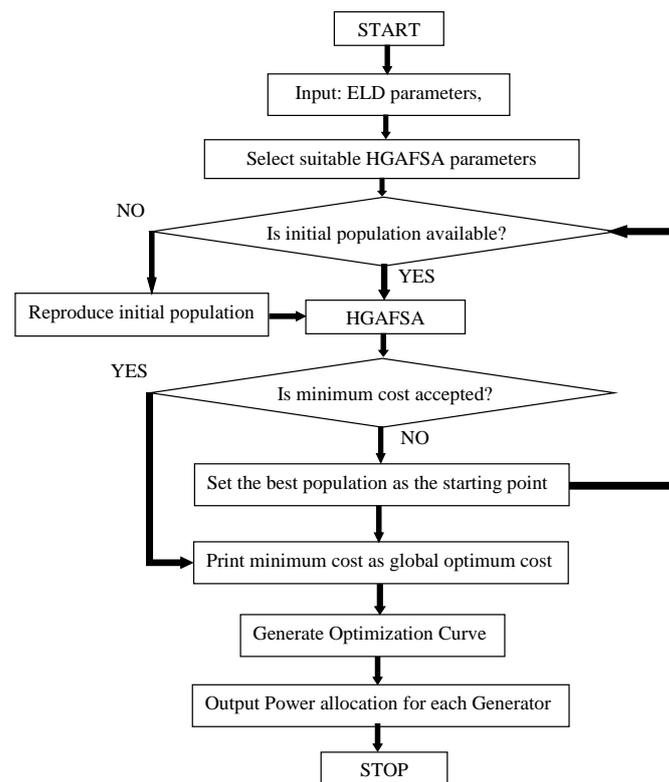


Figure 3. HGAFSA-based ELD algorithm.

4. Performance Validation

To demonstrate the effectiveness of the developed HGAFSA-based ELD problem solving algorithm, six test systems are designed. The developed algorithm is programmed in the MATLAB 2016a environment on a setup with 8 GB RAM and a 2.3 GHz Core I3 processing Computer running Windows 10.1.

A set of suitable parameters chosen for simulation analysis are listed in Table 2. To achieve the desired objective of outperforming all existing ELD algorithms in the literature, the best solution generated by HGAFSA is set as the starting point if it does not meet the desired goal. This process is repeated as far as the optimum cost is higher than the best cost presented in the literature so far.

Table 2. HGAFSA simulation parameter settings.

Parameter	Algorithm	Abbreviation	Value
Population Size	GA, AFSA	Psize	64
Number of Parameters	GA, AFSA	NoP	32
Visual Distance	AFSA	VD	0.875 to 1
Crowdness Factor	AFSA	CF	0.09 to 0.5
Step Size	AFSA	Ssize	0.00125 to 0.1
Max. Iteration	GA, AFSA	Max_Iter	10,000
Mutation Rate	GA	MR	0.4 to 0.75
Selection Probability	GA	SProb	0.375 to 0.5
Number of Bits	GA	NoB	8

4.1. Test System 1

The proposed HGAFSA algorithm is tested for a 13 generating units (Gen. units) test system having a non-smooth fuel cost function with valve-point effect, and its effectiveness is demonstrated compared to that of the oppositional grey wolf optimization (OGWO) algorithm [28] for the mentioned test system. The system data, such as fuel cost coefficients and active power limit thresholds for different generators, are adopted from [29] and the power demand assumed for the given test system is 2.52 GW. The simulation results of the 13-unit system are comparatively analyzed in Table 3 for our proposed HGAFSA and other optimization algorithms, such as improved coordinated aggregation-based PSO [30], shuffled differential evolution (SDE), oppositional real-coded chemical reaction optimization (ORCCRO) [31], biogeography-based optimization (BBO), hybrid differential evolution-based BBO (DE/BB) [32], and oppositional-invasive weed optimization (OIWO) [33]. The performance of HGAFSA is measured in terms of fuel cost (\$/h) and power loss (MW), compared to the mentioned optimization algorithms, in Table 3. The fuel cost of HGAFSA is lower (24,141.26 \$/h) compared to other algorithms. Comparatively, HGAFSA reduces the fuel cost of the generation system under consideration. Similarly, SDE and ORCCRO result in the same power loss (40.09 and 40.11) as HGAFSA (40.22), but lower than GWO, OGWO, and OIWO. Therefore, the reduced fuel cost and minimum power loss shown in Table 3 prove the efficacy of our proposed HGAFSA. Figure 4a shows the HGAFSA-based optimization curve. Figure 4a reveals that the algorithm converges at the thirteenth generation system to a cost of 24,141.2687 \$/h. This results in annual savings of \$3.253 m. Figure 4b presents the cumulative power generated by the units. In ELD, the power generated by each unit must lie within its maximum and the minimum allowable power output (upper and lower limits). Therefore, the power allocations must lie within this limit and the optimum power allocated to each unit (P_i) must lie within its limits. Finally, the cost functions of the various generating units influence the optimum power allocated the units and, thus, define the pattern of the 'optimum P_i ' curves for each of the six test systems in this work. In general, the peak of the cumulative power curves can be described as follows.

Table 3. Comparison of results for 13-unit system.

No. of Gen. Units	HGAFSA (MW)	OGWO (MW)	GWO (MW)	OIWO (MW)	SDE (MW)	ORCCRO (MW)
1	628.32	628.29	628.16	628.31	628.32	628.32
2	299.20	299.18	298.92	299.19	299.20	299.20
3	299.20	297.50	298.22	299.19	299.20	299.20
4	159.73	159.72	159.72	159.73	159.73	159.73
5	159.73	159.73	159.72	159.73	159.73	159.73
6	159.73	159.72	159.72	159.73	159.73	159.73
7	159.73	159.73	159.71	159.73	159.73	159.73
8	151.73	159.73	159.67	159.73	159.73	159.73
9	148.02	159.73	159.66	159.73	144.74	144.72
10	114.79	77.39	77.39	77.39	113.12	112.14
11	95.58	114.74	114.60	113.10	92.40	92.40
12	92.40	92.39	92.38	92.35	92.40	92.40
13	92.40	92.37	92.35	92.39	92.40	92.40
Fuel Cost (\$/h)	24,141.26	24,512.72	24,514.47	24,514.83	24,514.90	24,513.91
Power Loss (MW)	40.11	40.28	40.29	40.36	40.09	40.11

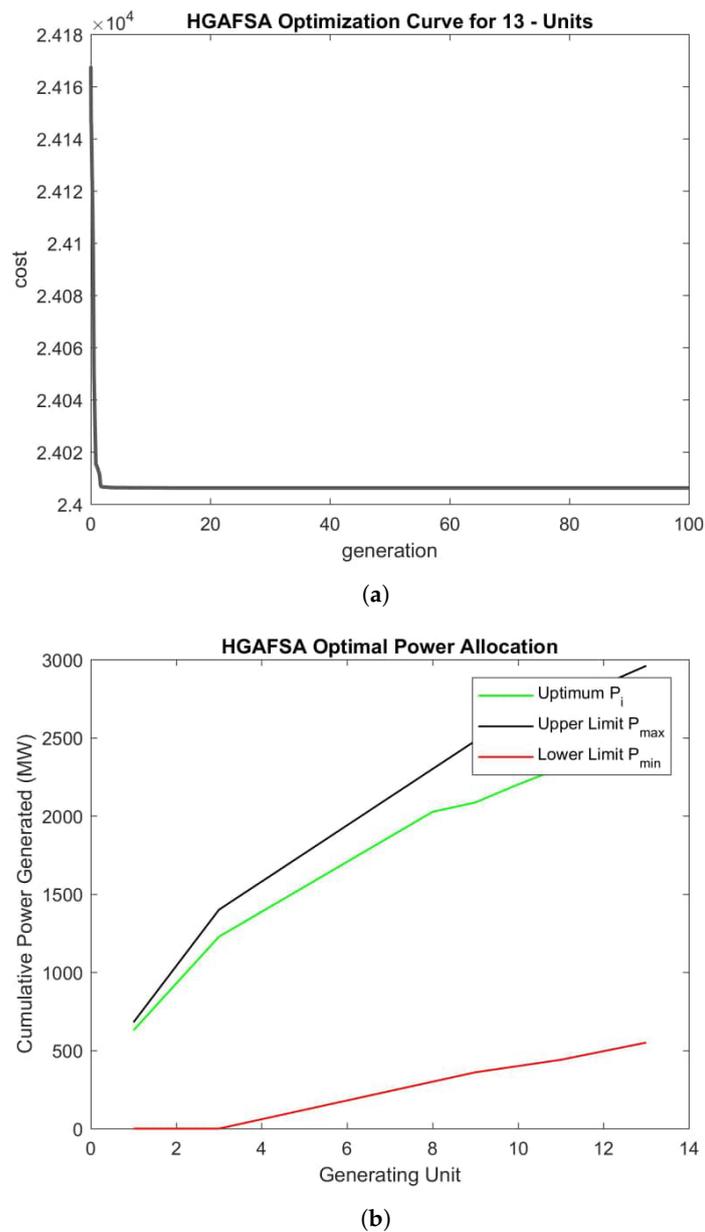


Figure 4. HGAFSA optimization and optimal power allocation curves for 13 units. (a) Optimization curve; (b) cumulative power generated.

1. Optimum P_i Curve: Defines the total power generated (Demand + Losses) by the system of generating units;
2. Upper Limit P_{max} Curve: Defines the maximum power that can be generated by the system of generating units;
3. Lower Limit P_{min} Curve: Defines the minimum power generated by the system of generating units.

In the case of Figure 4b, $P_i^{peak} = 25,604$, $P_{min}^{peak} = 550$, and $P_{max}^{peak} = 2960$.

4.2. Test System 2

The performance of the proposed algorithm, HGAFSA, and OGWO is validated for a 40 generating unit test system, where the data for valve-point coefficients and 40 generating units are adopted from [34]. A total power demand of 10.5GW is assumed for the given system. The real power generation output and fuel cost are calculated for the 40-unit

test system by various methods, such as HGAFSA, GWO, OGWO [28], SDE [31], ORC-CRO, quasi-oppositional teaching–learning-based optimization (QOTLBO) [35], hybrid ant colony–genetic algorithm (GA-API) [36], and krill herd algorithm (KHA) [37], and are tabulated in Table 4. The HGAFSA approach results in a reduced fuel cost (136,396.9 \$/h) and minimum power loss (957.29 MW) compared to the approaches mentioned in Table 4. Figure 5a shows the HGAFSA-based optimization curve, which converges at the fifty-first generation to a cost of 136,396.9727 \$/h. This results in annual savings of \$382,350. Furthermore, Figure 5b shows that all generators satisfy their inequality constraints.

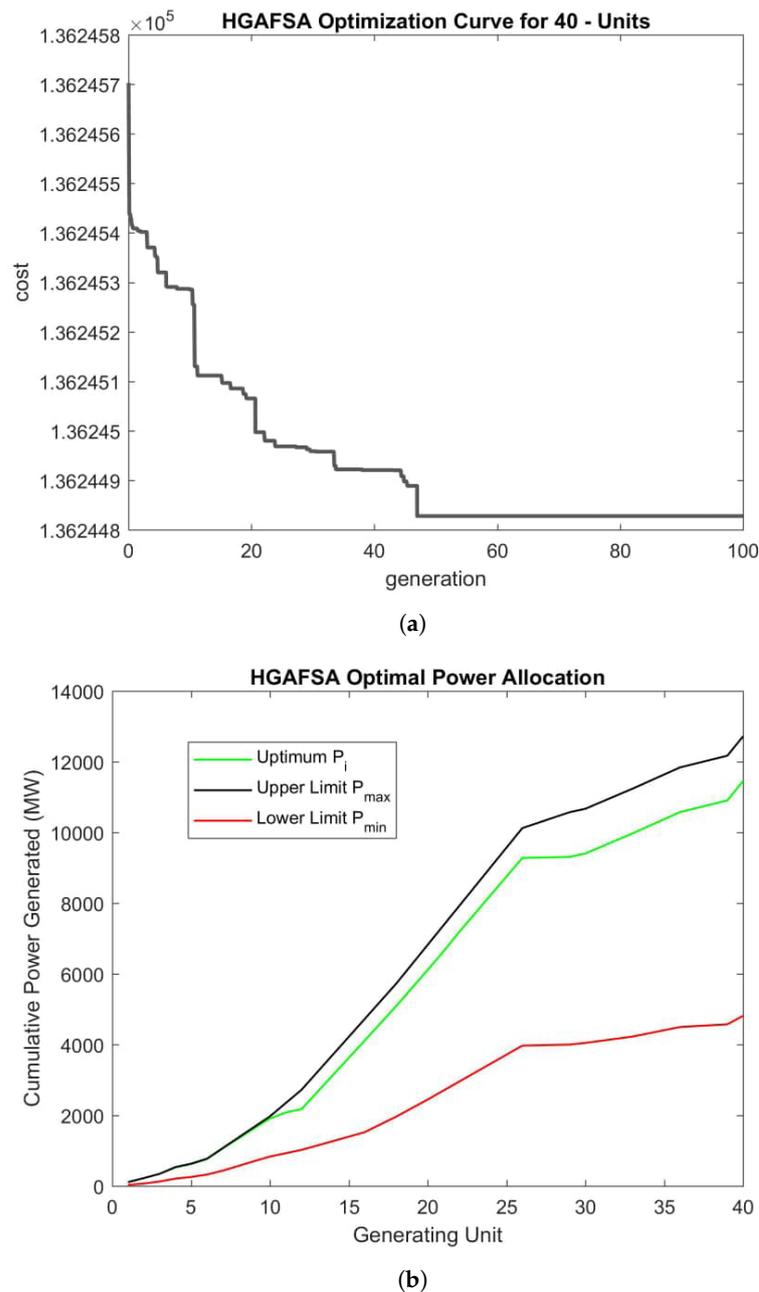


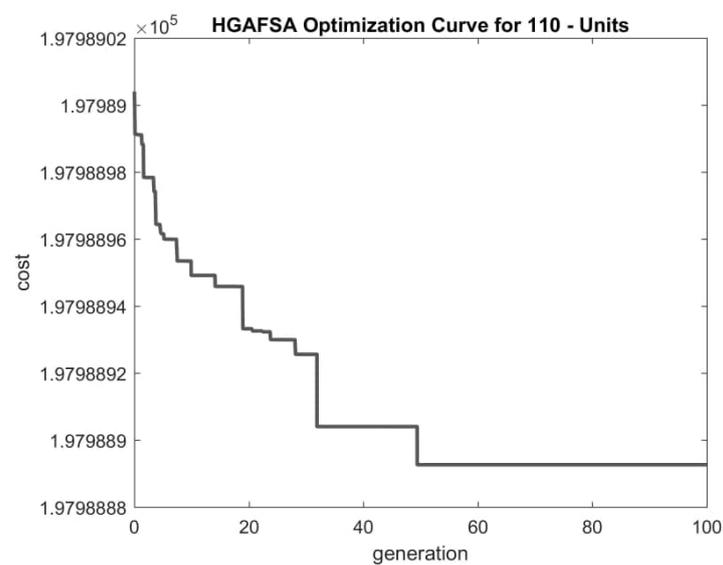
Figure 5. HGAFSA optimization and optimal power allocation curves for 40 units. (a) Optimization curve; (b) cumulative power generated.

Table 4. Comparison of results for 40-unit system.

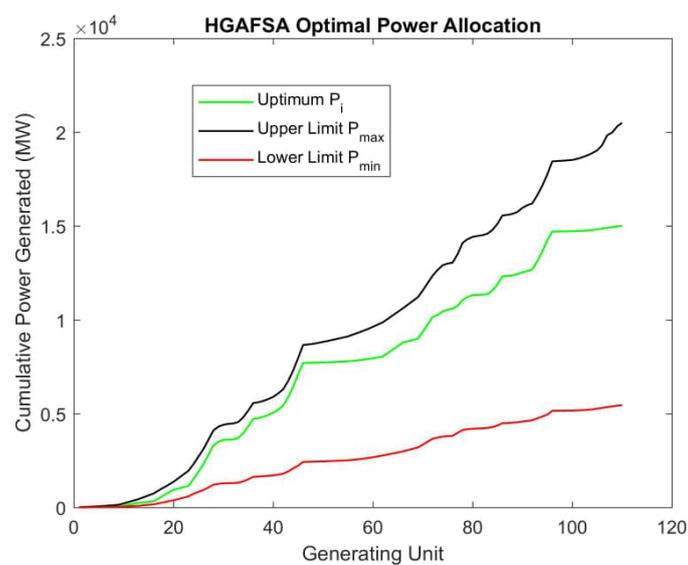
No. of Gen. Units	HGAFSA (MW)	OGWO (MW)	GWO (MW)	OIWO (MW)	SDE (MW)	ORCCRO (MW)	GAAPI (MW)	QOTLBO (MW)	KHA (MW)
1	113.96	114	114	113.9908	110.06	111.68	114	114	114
2	113.69	114	114	114	112.41	112.16	114	114	114
3	120	120	120	119.99	120	119.98	120	107.82	120
4	179.74	183.57	181.04	182.51	188.72	182.18	190	190	190
5	96.97	87.81	87.83	88.42	85.91	87.28	97	88.37	88.59
6	140	140	140	140	140	139.85	140	140	105.51
7	300	300	300	299.99	250.19	298.15	300	300	300
8	284.8	300	300	292.06	290.68	286.89	300	300	300
9	289.02	300	300	299.88	300	293.38	300	300	300
10	279.65	279.72	279.97	279.70	282.01	279.34	205.25	211.20	280.67
11	168.81	243.61	243.62	168.81	180.82	162.35	226.3	317.27	243.53
12	94	94.17	94.14	94	168.74	94.12	204.72	163.76	168.80
13	484.04	484.27	484.45	484.07	469.96	486.44	346.48	481.57	484.11
14	484.05	484.33	484.23	484.04	484.17	487.02	434.32	480.54	484.16
15	484.04	484.04	484.24	484.03	487.73	483.39	431.34	483.76	485.23
16	484.08	484.07	484.03	484.08	482.3	484.51	440.22	480.29	485.06
17	489.28	489.21	489.62	489.28	499.64	494.22	500	489.24	489.45
18	489.3	489.26	489.32	489.29	411.32	489.48	500	489.55	489.30
19	511.32	511.33	511.46	511.32	510.47	512.2	550	512.54	510.71
20	511.33	511.49	511.49	511.33	542.04	513.13	550	514.29	511.30
21	549.94	523.47	523.47	549.94	544.81	543.85	550	527.08	524.46
22	549.94	546.64	547.68	549.99	550	548	550	530.10	535.57
23	523.3	523.38	523.37	523.28	550	521.21	550	524.29	523.37
24	523.32	523.33	523.13	523.32	528.16	525.01	550	524.65	523.15
25	523.27	523.40	523.34	523.58	524.16	529.84	550	525.05	524.19
26	523.28	523.30	523.35	523.58	539.1	540.04	550	524.46	523.54
27	10.01	10.01	10.06	10.01	10	12.59	11.44	10.89	10.12
28	10.01	10.01	10.63	10.01	10.37	10.06	11.56	17.43	10.18
29	10.01	10.06	10.51	10.01	10	10.79	11.42	12.78	10.02
30	96.96	87.80	87.80	87.86	96.1	89.7	97	88.81	87.81
31	190	190	190	190	185.33	189.59	190	190	190
32	190	190	190	189.99	189.54	189.96	190	190	190
33	190	190	190	190	189.96	187.61	190	190	190
34	199.99	200	200	199.99	199.9	198.91	200	200	200
35	200	200	200	200	196.25	199.98	200	168.08	164.91
36	169.2	164.89	164.83	164.82	185.85	165.68	200	165.50	164.97
37	110	110	110	110	109.72	109.98	110	110	110
38	109.99	110	110	109.99	110	109.82	110	110	110
39	110	110	110	110	95.71	109.88	110	110	110
40	550	511.85	511.54	550	532.43	548.5	550	511.53	512.06
Fuel Cost (\$/h)	136,396.9	136,440.6	136,446.8	136,452.7	138,157	136,855.1	139,865	137,329.8	136,670
Power Loss (MW)	957.29	973.12	973.28	957.29	974.43	958.75	1045.06	1008.96	978.92

4.3. Test System 3

This system comprises 110 generating units with quadratic cost function characteristics. A load demand of 15GW is assumed for this system, and other system data (fuel valve coefficients and active power thresholds) are obtained from [38]. The minimum fuel cost and generation capacity computed using HGAFSA and OIWO are tabulated in Table 5. The results show that HGAFSA provides efficient and cheap power generation compared to OIWO and other optimization algorithms mentioned in the literature. Figure 6a shows the HGAFSA-based optimization curve. The algorithm is truncated at the hundredth generation and at a cost of 197,988.892 \$/h. This results in annual savings of \$2135.69. The saving in cost is lower because of the narrow margins for some of the generating units, especially units 1 to 9, and the high cost of generation. Furthermore, Figure 6b shows that all generators satisfy their inequality constraints.



(a)



(b)

Figure 6. HGAFSA optimization and optimal power allocation curves for 110 units. (a) Optimization curve; (b) cumulative power generated.

Table 5. Comparison of results for 110 unit system.

No. of Gen. Units	HGAFSA (MW)	OIWO (MW)	No. of Gen. Units	HGAFSA (MW)	OIWO (MW)	No. of Gen. units	HGAFSA (MW)	OIWO (MW)
1	2.4	2.4	38	69.99	69.98	75	89.99	89.99
2	2.40	2.40	39	99.99	99.99	76	49.99	49.99
3	2.40	2.40	40	120	120	77	160	160.01
4	2.4	2.4	41	157.18	156.8	78	295.76	291.36
5	2.4	2.4	42	220	220	79	175.05	177
6	4.01	4.01	43	440	440	80	98.01	97.75
7	4	4	44	560	560	81	10.01	10.01
8	4	4	45	660	660	82	12.01	12.30
9	4	4	46	616.43	619.53	83	20.01	20.04
10	64.39	63.05	47	5.40	5.40	84	199.98	199.99
11	62.16	59.27	48	5.4	5.4	85	324.99	324.51
12	36.29	35.65	49	8.40	8.40	86	439.99	439.99
13	56.62	57.43	50	8.4	8.4	87	14.42	18.86
14	25	25	51	8.4	8.4	88	24.32	23.33
15	25	25	52	12	12	89	82.44	84.40
16	25	25	53	12	12	90	89.25	91.9
17	155	155	54	12.01	12.01	91	57.61	58.29
18	155	155	55	12	12	92	99.99	98.07
19	155	155	56	25.2	25.2	93	440	440
20	155	155	57	25.2	25.2	94	499.99	499.97
21	68.9	68.9	58	35	35	95	600	600
22	68.9	68.9	59	35.01	35	96	471.47	469.27
23	68.9	68.9	60	45.01	45.01	97	3.6	3.6
24	350	350	61	45.01	45.01	98	3.6	3.6
25	400	400	62	45	45	99	4.4	4.4
26	400	400	63	184.99	185	100	4.40	4.40
27	500	500	64	185	184.99	101	10.01	10.01
28	500	500	65	185	185	102	10.01	10.01
29	200	199.99	66	184.99	185	103	20.01	20.01
30	100	100	67	70	70	104	20.01	20.01
31	10.01	10.01	68	70	70	105	40	40
32	19.99	19.99	69	70.01	70.01	106	40.01	40.01
33	79.99	79.48	70	359.99	360	107	50	50
34	250	250	71	400	400	108	30	30
35	360	360	72	400	400	109	40	40
36	400	399.99	73	104.96	107.83	110	20	20
37	39.99	39.99	74	191.49	188.81	Fuel Cost (\$/h)	197,988.8	197,989.1

4.4. Test System 4

In this scenario, a power system comprising 140 generating units is considered and the performance of the proposed HGAFSA algorithm is compared to OGWO, SDE, and OIWA. These results validate that HGAFSA outperforms all other methods in producing cheap power. The performance of both HGAFSA and OGWO is satisfactory but HGAFSA is significantly better than OGWO. The fuel costs are computed as 1,558,619 \$/h for HGAFSA and 1,559,710 \$/h and 1,559,953 \$/h for OGWO and GWO, respectively. Figure 7a shows the HGAFSA-based optimization curve. HGAFSA converges at the sixty-sixth generation to a

cost of 1,559,710 \$/h, resulting in annual savings of \$9.55 m. Furthermore, Figure 7b shows that all generators satisfy their inequality constraints.

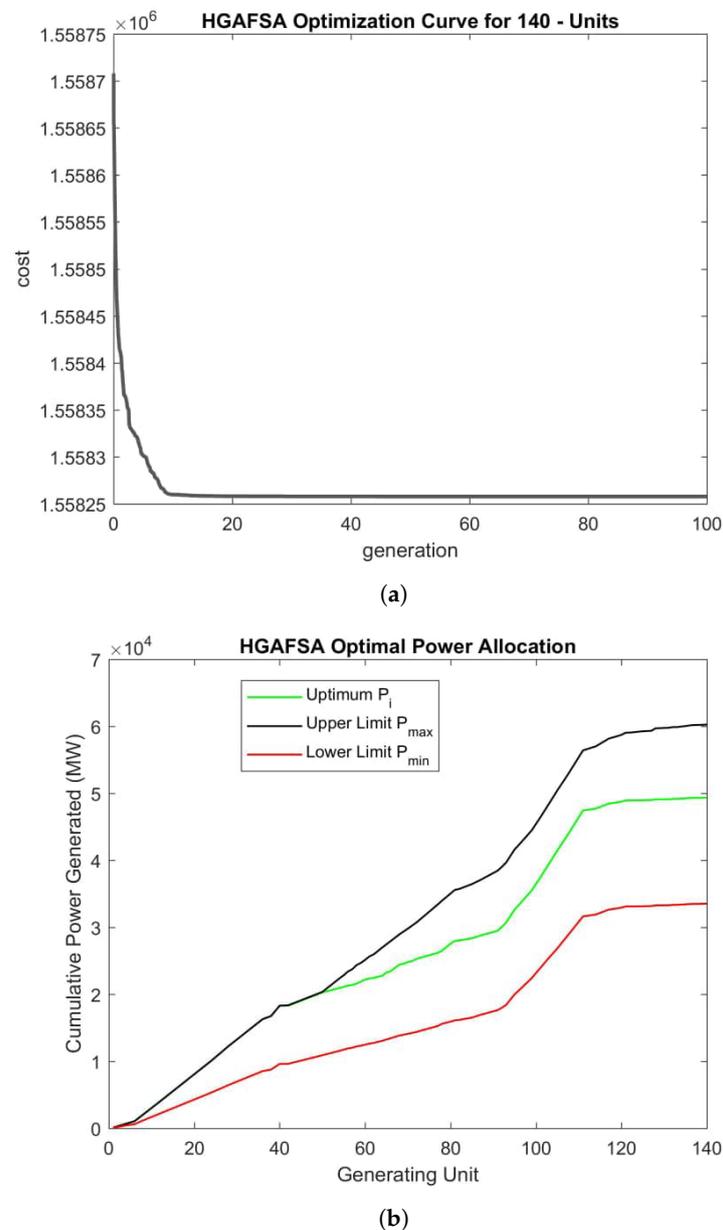
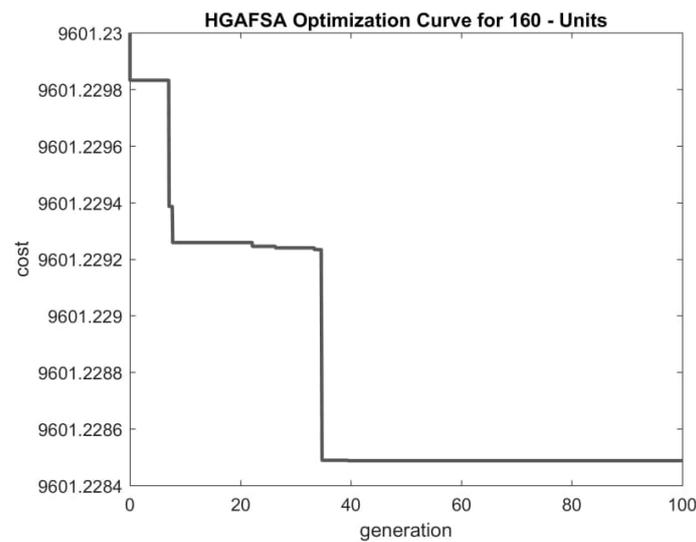


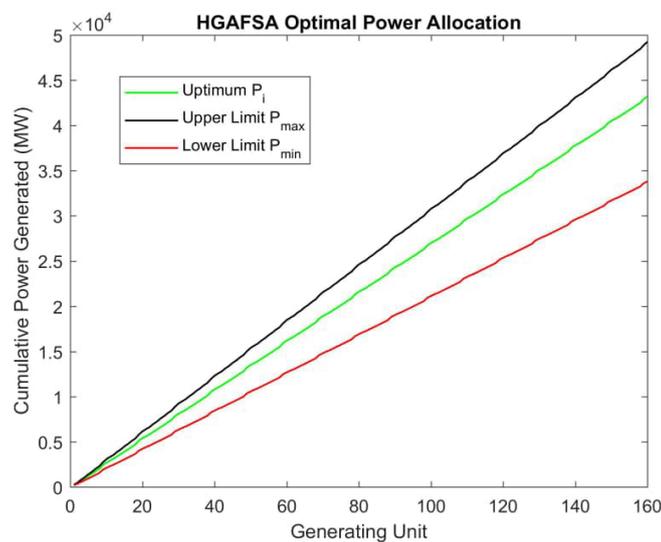
Figure 7. HGAFSA optimization and optimal power allocation curves for 140 units. (a) Optimization curve; (b) cumulative power generated.

4.5. Test System 5

The proposed algorithm is also tested for a test system consisting of 160 generating units with non-smooth valve-point cost functions. In order to show the effectiveness of the proposed technique for solving a large-scale ELD problem, transmission losses are ignored. The fuel costs are computed as 9612.8 \$/h for HGAFSA and 9745.1 \$/h and 9813.3 \$/h for OGWO and GWO, respectively. These results indicate that the total production cost is lower than that of all other methods mentioned in this work. Figure 8a shows the HGAFSA-based optimization curve. The HGAFSA algorithm converges at the eighth generation to a cost of 9612.8295 \$/h, providing annual savings of \$1.158 m. Furthermore, Figure 8b shows that all generators satisfy their inequality constraints.



(a)



(b)

Figure 8. HGAFSA optimization and optimal power allocation curves for 160 units. (a) Optimization curve; (b) cumulative power generated.

4.6. Test System 6

The test system comprises a 463-unit system formed by combining test systems 1 to 5 (Gen. Units of 13 + 40 + 110 + 140 + 160). The coefficient of the valve point is assumed to be zero if not available for any generating unit. The simulation study is carried out under valve-point loading multi-fuel cost and emotion. A total power demand of 120,000 MW is used for the dispatch problem. The power loss components are assumed to be negligible. The problem is solved using the proposed HGAFSA. The fuel cost is computed as 1,645,338.7 \$/h. Figure 9a also shows the HGAFSA-based optimization curve. The HGAFSA algorithm converges at the eighty-sixth generation to a cost of 1,645,338.7385 \$/h. Furthermore, Figure 9b shows that all generators satisfy their inequality constraints.

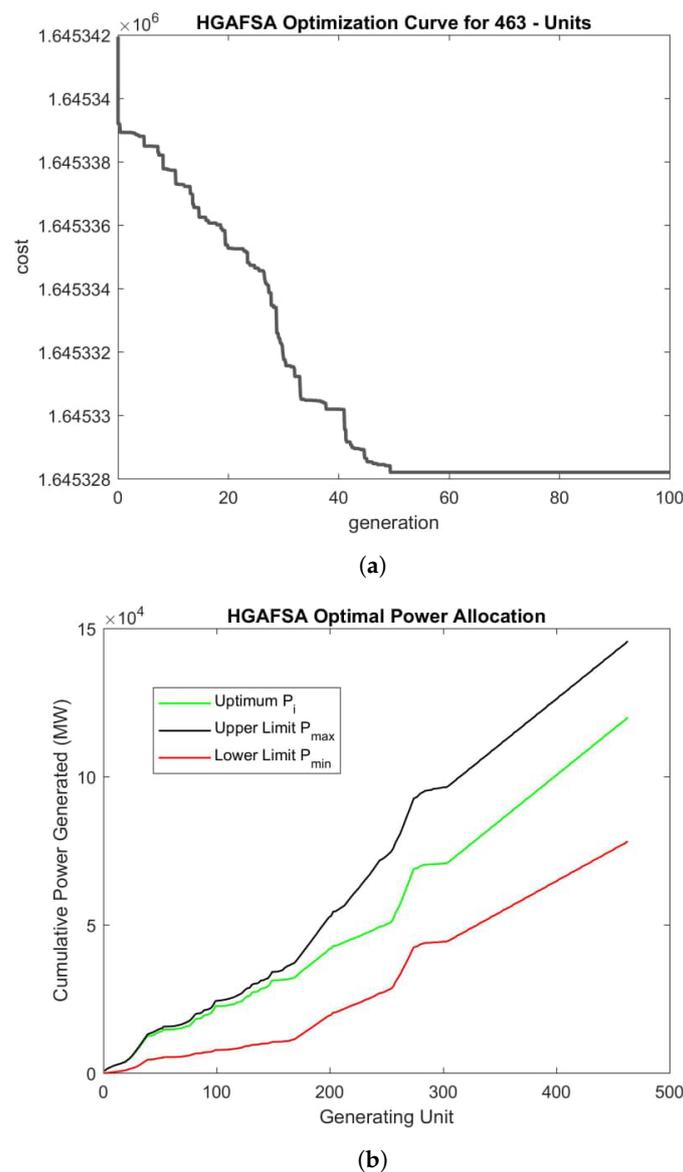


Figure 9. HGAFSA optimization and optimal power allocation curves for 463 units. (a) Optimization curve; (b) cumulative power generated.

In general, the results of the overall simulation study are summarized and presented in Table 6, which provides a comparative analysis of the developed HGAFSA and the best-performing algorithms mentioned in the literature. Table 6 reveals that HGAFSA outperforms all other algorithms for all simulation scenarios, with the best performance obtained in the 140-unit test system. This, however, results in relatively high overall annual savings. This work also provides a new benchmark test system of 463 units. In order to further demonstrate the optimization parameter sensitivity of the proposed HGAFSA, a range of sensitivity analysis is performed and is reported in the next subsection.

4.7. Sensitivity Analysis

Sensitivity analysis is carried out to provide an insight towards the selection of the HGAFSA-based optimization parameter settings (a) to achieve the desired optimization trade-off, (b) to boost the capability of replicating this work, and (c) to ease future advancement in this area of research. Tables 7–13 present the optimization results obtained by varying the most sensitive HGAFSA parameters over a number of Monte Carlo (NMC) simulation trials. In the analysis, the NMC is maintained at 10 trials and the optimization

result is averaged over the 10 trials. In each of the tables, the convergence optimum cost, the number of function evaluations before convergence (NFEBC), the number of generations/iterations before convergence (NGBC), and the CPU time are averaged (over the 10 trials/NMC) and recorded/presented.

The convergence/optimization curve reaches a minimum value at a point referred to as the 'point of convergence'. The number of optimization trials/generation before the point of convergence, also known as the number of generations before convergence (NGBC), is used to compare the performance of HGAFSA for different cases of optimization. Contrarily, the NFEBC gives an idea of the computational intensity/expensiveness of any given optimization algorithm. In Table 7, the Psize is randomly varied between 2 and 500 individual/candidate solutions, and it is observed that the best performance (i.e., the lowest cost and a relatively lower NFEBC, NGBC, and CPU time) is achieved with 64-candidate solutions. Furthermore, for Psize greater than 64-candidate solutions, HGAFSA yields the optimum cost but with higher NFEBC and CPU time. However, the NGBC is observed to decrease with an increase in Psize.

Table 6. Summary of results for the comparison of HGAFSA and OGWO/OIWO.

Test System	Number of Gen. Units	HGAFSA (\$)	Best Cost in Literature (\$)	Annual Savings (\$)	Total Power (MW)	CPU (s)
1	13	24,141.26	24,512.72	3,253,957.2	2560.36	5.02
2	40	136,396.97	136,440.62	382,350.35	11,457.29	10.11
3	110	197,988.89	197,989.14	2135.68	15,000	104.3
4	140	1,558,619.09	1,559,710	9,556,337	49,342	47.12
5	160	9612.82	9745.11	1,158,803.5	43,200	10.23
6	463	1,645,338.73	none	none	120,000	

Table 7. Effect of population size (Psize) on HGAFSA convergence.

NoP = 32; VD = 1; CF = 0.5; Ssize = 0.005; Max.Iter = 100; MR = 0.5; Sprob = 0.5; NoB = 8; NMC = 10					
Test Sys.	Psize	Cost (\$)	NFEBC	NGBC	CPU (s)
1	2	1,558,959	44,544	87	5.823
2	16	1,558,651	319,488	78	41.77
3	64	1,558,619	360,448	22	47.12
4	100	1,558,619	460,800	18	60.24
5	500	1,558,619	1,536,000	12	200.8

Table 8. Effect of number of parameters (NoP) on HGAFSA convergence.

Psize = 64; VD = 1; CF = 0.5; Ssize = 0.005; Max.Iter = 100; MR = 0.5; Sprob = 0.5; NoB = 8; NMC = 10					
Test Sys.	NoP	Cost (\$)	NFEBC	NGBC	CPU (s)
1	2	1,559,930	51,200	100	6.693
2	16	1,558,700	729,088	89	95.31
3	32	1,558,619	360,448	22	47.12
4	64	1,558,619	622,592	19	81.39
5	124	1,558,619	1,079,296	17	141.1

Table 9. Effect of visual distance (VD) on HGAFSA convergence.

Psize = 64; NoP = 32; CF = 0.5; Ssize = 0.005; Max.Iter = 100; MR = 0.5; Sprob = 0.5; NoB = 8; NMC = 10					
Test Sys.	VD	Cost (\$)	NFEBc	NGBC	CPU (s)
1	0.25	1,559,130	1,638,400	100	71.39
2	0.5	1,559,009	1,638,400	100	71.39
3	0.75	1,558,909	1,490,944	91	64.97
4	1	1,558,619	360,448	22	47.12
5	1.25	1,558,619	491,520	30	50.25
6	1.2	1,559,658	1,638,400	100	71.39

Table 10. Effect of crowdedness factor (CF) on HGAFSA convergence.

Psize = 64; NoP = 32; VD = 0.875; Ssize = 0.005; Max.Iter = 100; MR = 0.5; Sprob = 0.5; NoB = 8; NMC = 10					
Test Sys.	CF	Cost (\$)	NFEBc	NGBC	CPU (s)
1	0.03	1,559,992	1,638,400	100	71.39
2	0.06	1,559,986	1,556,480	95	67.82
3	0.09	1,559,930	1,490,944	91	64.97
4	0.5	1,558,619	360,448	22	47.12
5	0.6	1,558,666	851,968	52	55.69
6	0.7	1,558,802	1,343,488	82	58.54

Table 11. Effect of step size (Ssize) on HGAFSA convergence.

Psize = 64; NoP = 32; VD = 0.875; CF = 0.5; Max.Iter = 100; MR = 0.5; Sprob = 0.5; NoB = 8; NMC = 10					
Test Sys.	Ssize	Cost (\$)	NFEBc	NGBC	CPU (s)
1	0.00125	1,558,619	360,448	22	70.68
2	0.005	1,558,619	360,448	22	47.12
3	0.01	1,558,689	524,288	32	68.54
4	0.025	1,558,679	1,097,728	67	71.75
5	0.05	1,558,692	1,343,488	82	58.54
6	0.1	1,558,629	1,507,328	92	65.68

Step size (Ssize) is another parameter of the HGAFSA that randomly affects its characteristics. In general, the smaller the value of Ssize, the better the performance, as presented in Table 11. An Ssize of 0.005 is used in this work. An important parameter of the HGAFSA is the selection probability (Sprob). Sprob also randomly affects the characteristics of HGAFSA, as shown in Table 12. Maintaining the value of Sprob at 50% (0.5) can work for most optimization problems. Similar to Sprob, the mutation rate (MR) also randomly affects the behavior of the developed HGAFSA, as shown in Table 13. A 50% mutation rate (MR = 0.5) is sufficient to handle most optimization problems. Finally, it is noted that, despite the effect of parameter variation, the HGAFSA outperforms most of the existing algorithms, regardless of its parameter settings.

Table 12. Effect of selection probability (Sprob) on HGAFSA convergence.

Psize = 64; NoP = 32; VD = 0.875; CF = 0.5; Max.Iter = 100; MR = 0.5; Ssize = 0.005; NoB = 8; NMC = 10					
Test Sys.	Sprob	Cost (\$)	NFEBBC	NGBC	CPU (s)
1	0.1	1,558,629	442,368	27	48.19
2	0.2	1,558,621	507,904	31	55.33
3	0.3	1,558,650	475,136	29	51.76
4	0.5	1,558,619	360,448	22	47.12
5	0.7	1,558,629	622,592	38	67.82
6	0.8	1,558,622	491,520	30	64.25

Table 8 shows that an increase in the number of parameters (NoP) results in earlier convergence but with higher computational expensiveness and convergence time. It is also found out that a choice of 32 parameters is sufficient to achieve a relatively good convergence characteristic. Table 9 presents the effect of the visual distance (VD) on the convergence characteristics of HGAFSA. The choice of VD alters the value of NGBC, and an improper selection of VD may result in a large value of NGBC or even lead to local minimum/premature convergence. A trail and error procedure can be used to obtain a suitable VD for any given optimization problem. However, it is found that a VD = 1 is most suitable for solving the ELD problem. The effect of the crowdedness factor (CF) is demonstrated in Table 10. Even though the results reported are the average of 10 trials, the effect of CF on HGAFSA is observed to be random. However, with CF = 0.5, the least/optimum cost is obtained for all scenarios.

Table 13. Effect of mutation rate (MR) on HGAFSA convergence.

Psize = 64; NoP = 32; VD = 0.875; CF = 0.5; Max.Iter = 100; Ssize = 0.005; Sprob = 0.5; NoB = 8; NMC = 10					
Test System	MR	Cost (\$)	NFEBBC	NGBC	CPU (s)
1	0.1	1,558,629	540,672	33	58.9
2	0.25	1,558,621	638,976	39	69.61
3	0.5	1,558,619	360,448	22	47.12
4	0.75	1,558,650	524,288	32	57.12
5	1	1,558,629	573,440	35	62.47

5. Conclusions and Future Directions

HGAFSA is developed using a hybridization of the conventional binary-coded GA and real-coded AFSA using a decoder (which converts a GA 'chromosome' into an AFSA 'fish') and an encoder (which performs the reverse operation of the decoder). An ELD encoder algorithm is also developed and integrated with the HGAFSA to form a more robust, efficient, and reliable technique (with a guaranteed high rate of convergence) for solving complex ELD problems. A sensitivity analysis is carried out on the optimization parameter settings of the developed HGAFSA. It is concluded that a trial and error method can be used to select a suitable set of parameters in order to optimize an ELD problem. The convergence of HGAFSA is randomly dependent on most of its parameters, and despite the limitation of the parameter choice, its solutions would always be better than most of the existing algorithms. The choice of encoder/decoder function further enhances the optimization process. This feature of HGAFSA makes it robust and almost insensitive to the choice of optimization parameter setting and enables it to support a wide range

of optimization parameter settings. The effectiveness of the HGAFSA is demonstrated through the simulation of solutions to a multi-objective ELD problem with higher-order cost functions using 13, 40, 110, 140, 160, and 463 generating unit test systems. Outstanding performance is demonstrated by the proposed HGAFSA-based ELD approach for most of the test systems. The future directions include the integration of the proposed ELD encoder with other meta-heuristics techniques in order to increase the efficiency and performance and its comparison with the proposed HGAFSA. The computational expensiveness of HGAFSA can also be further reduced by minimizing the number of repetitive steps in the algorithm while still introducing a powerful function that could compensate for their solution search ability. Another possible improvement is to modify the solution search procedure by introducing a set of steps that could adjust the parameter settings of HGAFSA based on the iteration counter to further enhance its rate of convergence.

Author Contributions: Conceptualization, A.M.K., M.K. and F.A.; methodology, F.A. and F.M.; formal analysis, M.K., F.A. and F.M.; validation, A.M.K. and F.A.; investigation, F.A. and Z.U.; funding acquisition, F.R.A. and G.H.; resources, F.R.A. and G.H.; data curation, M.K. and F.M.; project administration, Z.U., F.R.A. and G.H.; writing—original draft preparation, A.M.K., and M.K.; writing—review and editing, Z.U., F.R.A. and G.H.; supervision, F.M., F.A. All authors have read and agreed to the published version of the manuscript.

Funding: The APC is funded by Taif University Researchers Supporting Project Number(TURSP-2020/331), Taif University, Taif, Saudi Arabia.

Acknowledgments: The authors would like to acknowledge the support from Taif University Researchers Supporting Project Number(TURSP-2020/331), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Li, Q.; Wang, J.; Zhang, Y.; Fan, Y.; Bao, G.; Wang, X. Multi-Period Generation Expansion Planning for Sustainable Power Systems to Maximize the Utilization of Renewable Energy Sources. *Sustainability* **2020**, *12*, 1083. [\[CrossRef\]](#)
- Chouhdry, Z.U.R.; Hasan, K.M.; Raja, M.A.Z. Design of reduced search space strategy based on integration of Nelder–Mead method and pattern search algorithm with application to economic load dispatch problem. *Neural Comput. Appl.* **2018**, *30*, 3693–3705. [\[CrossRef\]](#)
- Zhang, Q.; Zou, D.; Duan, N.; Shen, X. An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem. *Appl. Soft Comput.* **2019**, *78*, 641–669. [\[CrossRef\]](#)
- Babatunde, O.; Munda, J.; Hamam, Y. Power system flexibility: A review. *Energy Rep.* **2020**, *6*, 101–106.
- Azad, M.A.K.; Rocha, A.M.A.; Fernandes, E.M. Improved binary artificial fish swarm algorithm for the 0–1 multidimensional knapsack problems. *Swarm Evol. Comput.* **2014**, *14*, 66–75. [\[CrossRef\]](#)
- Iqbal, S.; Sarfraz, M.; Ayyub, M.; Tariq, M.; Chakraborty, R.K.; Ryan, M.J.; Alamri, B. A Comprehensive Review on Residential Demand Side Management Strategies in Smart Grid Environment. *Sustainability* **2021**, *13*, 7170. [\[CrossRef\]](#)
- Mehmood, F.; Ashraf, N.; Alvarez, L.; Malik, T.N.; Qureshi, H.K.; Kamal, T. Grid integrated photovoltaic system with fuzzy based maximum power point tracking control along with harmonic elimination. *Trans. Emerg. Telecommun. Technol.* **2020**, e3856. [\[CrossRef\]](#)
- Ellahi, M.; Abbas, G.; Satrya, G.B.; Usman, M.R.; Gu, J. A Modified Hybrid Particle Swarm Optimization With Bat Algorithm Parameter Inspired Acceleration Coefficients for Solving Eco-Friendly and Economic Dispatch Problems. *IEEE Access* **2021**, *9*, 82169–82187. [\[CrossRef\]](#)
- Bernardon, D.; Mello, A.; Pfitscher, L.; Canha, L.; Abaide, A.; Ferreira, A. Real-time reconfiguration of distribution network with distributed generation. *Electr. Power Syst. Res.* **2014**, *107*, 59–67. [\[CrossRef\]](#)
- Atwa, Y.; El-Saadany, E.; Salama, M.; Seethapathy, R. Optimal renewable resources mix for distribution system energy loss minimization. *IEEE Trans. Power Syst.* **2010**, *25*, 360–370. [\[CrossRef\]](#)
- Mahmoud, K.; Abdel-Nasser, M.; Mustafa, E.; Ali, Z.M. Improved Salp–Swarm Optimizer and Accurate Forecasting Model for Dynamic Economic Dispatch in Sustainable Power Systems. *Sustainability* **2020**, *12*, 576. [\[CrossRef\]](#)
- Aziz, A.S.; Tajuddin, M.F.N.; Adzman, M.R.; Ramli, M.A.M.; Mekhilef, S. Energy Management and Optimization of a PV/Diesel/Battery Hybrid Energy System Using a Combined Dispatch Strategy. *Sustainability* **2019**, *11*, 683. [\[CrossRef\]](#)
- Deb, S.; Abdelminaam, D.S.; Said, M.; Houssein, E.H. Recent Methodology-Based Gradient-Based Optimizer for Economic Load Dispatch Problem. *IEEE Access* **2021**, *9*, 44322–44338. [\[CrossRef\]](#)
- Alanazi, M.S. A Modified Teaching–Learning-Based Optimization for Dynamic Economic Load Dispatch Considering Both Wind Power and Load Demand Uncertainties With Operational Constraints. *IEEE Access* **2021**, *9*, 101665–101680. [\[CrossRef\]](#)

15. Dhebar, Y.; Deb, K. A computationally fast multimodal optimization with push enabled genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*; ACM: New York, NY, USA, 2017; pp. 191–192.
16. Fang, N.; Zhou, J.; Zhang, R.; Liu, Y.; Zhang, Y. A hybrid of real coded genetic algorithm and artificial fish swarm algorithm for short-term optimal hydrothermal scheduling. *Int. J. Electr. Power Energy Syst.* **2014**, *62*, 617–629. [[CrossRef](#)]
17. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
18. Jubril, A.M.; Komolafe, O.A.; Alawode, K.O. Solving multi-objective economic dispatch problem via semidefinite programming. *IEEE Trans. Power Syst.* **2013**, *28*, 2056–2064. [[CrossRef](#)]
19. Dharamjit, D. Load flow analysis on IEEE 30 bus system. *Int. J. Sci. Res. Publ.* **2012**, *2*, 1–6.
20. Sivagnanam, G.; Mauryan, K.C.; Maheswar, R.; Subramanian, R. An Efficient ITA Algorithm to Solve Economic Load Dispatch with Valve Point Loading Effects. *Int. J. Print. Packag. Allied Sci.* **2017**, *2017*, 121–128.
21. Khoa, T.; Vasant, P.; Singh, B.; Dieu, V. Solving Economic Dispatch By Using Swarm Based Mean-Variance Mapping Optimization (MVMOS). *Glob. Technol Optim* **2015**, *6*, 184.
22. Al-Bahrani, L.; Seyedmahmoudian, M.; Horan, B.; Stojcevski, A. Solving the Real Power Limitations in the Dynamic Economic Dispatch of Large-Scale Thermal Power Units under the Effects of Valve-Point Loading and Ramp-Rate Limitations. *Sustainability* **2021**, *13*, 1274. [[CrossRef](#)]
23. Jiang, C.; Wan, L.; Sun, Y.; Li, Y. The application of PSO-AFSA method in parameter optimization for underactuated autonomous underwater vehicle control. *Math. Probl. Eng.* **2017**, *2017*, 7482. [[CrossRef](#)]
24. Costa, M.F.P.; Rocha, A.M.A.; Fernandes, E.M. An artificial fish swarm algorithm based hyperbolic augmented Lagrangian method. *J. Comput. Appl. Math.* **2014**, *259*, 868–876. [[CrossRef](#)]
25. Hussain, I.; Ali, S.; Khan, B.; Ullah, Z.; Mehmood, C.; Jawad, M.; Farid, U.; Haider, A. Stochastic wind energy management model within smart grid framework: A joint Bi-directional service level agreement (SLA) between smart grid and wind energy district prosumers. *Renew. Energy* **2019**, *134*, 1017–1033. [[CrossRef](#)]
26. Ali, S.M.; Ullah, Z.; Mokryani, G.; Khan, B.; Hussain, I.; Mehmood, C.A.; Farid, U.; Jawad, M. Smart grid and energy district mutual interactions with demand response programs. *IET Energy Syst. Integr.* **2020**, *2*, 1–8. [[CrossRef](#)]
27. Chatterjee, S.; Laudato, M.; Lynch, L.A. Genetic algorithms and their statistical applications: An introduction. *Comput. Stat. Data Anal.* **1996**, *22*, 633–651. [[CrossRef](#)]
28. Li, M.; Huang, X.; Liu, H.; Liu, B.; Wu, Y.; Xiong, A.; Dong, T. Prediction of gas solubility in polymers by back propagation artificial neural network based on self-adaptive particle swarm optimization algorithm and chaos theory. *Fluid Phase Equilibria* **2013**, *356*, 11–17. [[CrossRef](#)]
29. Bhattacharya, A.; Chattopadhyay, P.K. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Trans. Power Syst.* **2010**, *25*, 1955–1964. [[CrossRef](#)]
30. Vlachogiannis, J.G.; Lee, K.Y. Economic load dispatch—A comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO. *IEEE Trans. Power Syst.* **2009**, *24*, 991–1001. [[CrossRef](#)]
31. Reddy, A.S.; Vaisakh, K. Shuffled differential evolution for large scale economic dispatch. *Electr. Power Syst. Res.* **2013**, *96*, 237–245. [[CrossRef](#)]
32. Ellahi, M.; Abbas, G. A Hybrid Metaheuristic Approach for the Solution of Renewables-Incorporated Economic Dispatch Problems. *IEEE Access* **2020**, *8*, 127608–127621. [[CrossRef](#)]
33. Roy, P.K.; Bhui, S. Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem. *Int. J. Electr. Power Energy Syst.* **2013**, *53*, 937–948. [[CrossRef](#)]
34. Mandal, B.; Roy, P.K.; Mandal, S. Economic load dispatch using krill herd algorithm. *Int. J. Electr. Power Energy Syst.* **2014**, *57*, 1–10. [[CrossRef](#)]
35. Wang, Y.; Li, B.; Weise, T. Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems. *Inf. Sci.* **2010**, *180*, 2405–2420. [[CrossRef](#)]
36. Chiang, C.L. Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. *IEEE Trans. Power Syst.* **2005**, *20*, 1690–1699. [[CrossRef](#)]
37. Muralikrishnan, N.; Jebaraj, L.; Rajan, C.C.A. A Comprehensive Review on Evolutionary Optimization Techniques Applied for Unit Commitment Problem. *IEEE Access* **2020**, *8*, 132980–133014. [[CrossRef](#)]
38. Vishwakarma, K.K.; Dubey, H.M.; Pandit, M.; Panigrahi, B.K. Simulated annealing approach for solving economic load dispatch problems with valve point loading effects. *Int. J. Eng. Sci. Technol.* **2012**, *4*, 60–72. [[CrossRef](#)]