

Article

An Innovative Cloud-Fog-Based Smart Grid Scheme for Efficient Resource Utilization

Fahad Alsokhry ^{1,2}, Andres Annuk ^{3,*}, Mohamed A. Mohamed ^{4,*} and Manoel Marinho ⁵

- ¹ Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
- ² K. A. CARE Energy Research and Innovation Center, King Abdulaziz University, Jeddah 21589, Saudi Arabia
- ³ Institute of Forestry and Engineering, Estonian University of Life Sciences, 51006 Tartu, Estonia
- ⁴ Electrical Engineering Department, Faculty of Engineering, Minia University, Minia 61519, Egypt
- ⁵ Polytechnic School of Pernambuco, University of Pernambuco (UPE), Recife 50720-001, PE, Brazil
- * Correspondence: andres.annuk@emu.ee (A.A.); dr.mohamed.abdelaziz@mu.edu.eg (M.A.M.)

Abstract: Smart grids (SGs) enhance the effectiveness, reliability, resilience, and energy-efficient operation of electrical networks. Nonetheless, SGs suffer from big data transactions which limit their capabilities and can cause delays in the optimal operation and management tasks. Therefore, it is clear that a fast and reliable architecture is needed to make big data management in SGs more efficient. This paper assesses the optimal operation of the SGs using cloud computing (CC), fog computing, and resource allocation to enhance the management problem. Technically, big data management makes SG more efficient if cloud and fog computing (CFC) are integrated. The integration of fog computing (FC) with CC minimizes cloud burden and maximizes resource allocation. There are three key features for the proposed fog layer: awareness of position, short latency, and mobility. Moreover, a CFC-driven framework is proposed to manage data among different agents. In order to make the system more efficient, FC allocates virtual machines (VMs) according to load-balancing techniques. In addition, the present study proposes a hybrid gray wolf differential evolution optimization algorithm (HGWDE) that brings gray wolf optimization (GWO) and improved differential evolution (IDE) together. Simulation results conducted in MATLAB verify the efficiency of the suggested algorithm according to the high data transaction and computational time. According to the results, the response time of HGWDE is 54 ms, 82.1 ms, and 81.6 ms faster than particle swarm optimization (PSO), differential evolution (DE), and GWO. HGWDE's processing time is 53 ms, 81.2 ms, and 80.6 ms faster than PSO, DE, and GWO. Although GWO is a bit more efficient than HGWDE, the difference is not very significant.

Keywords: cloud computing; fog computing; improved differential evolution; gray wolf optimization; efficient resource utilization; smart grid



Citation: Alsokhry, F.; Annuk, A.; Mohamed, M.A.; Marinho, M. An Innovative Cloud-Fog-Based Smart Grid Scheme for Efficient Resource Utilization. *Sensors* **2023**, *23*, 1752. <https://doi.org/10.3390/s23041752>

Academic Editors: Juan M. Corchado, Hossam A. Gabbar and Hsiao-Chun Wu

Received: 12 October 2022
Revised: 12 January 2023
Accepted: 2 February 2023
Published: 4 February 2023



Copyright: © 2023 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

Smart grids (SG) are essential components of smart communities. A fast and efficient communication system to monitor and control power consumption is used in these grids to establish a smart and distributed system that can control power needs economically and sustainably [1]. In this regard, the Internet of Energy (IoE) could be thought of as a combination of SG and the Internet of Things (IoT) [2]. IoE aims to simplify the power trade among users by providing an effective structure. Internet controls and observes intermittent and distributed power production and storage logically. A wide variety of loads and sources will be allowed to trade power through the IoE, such as distributed energy storage, renewable energy resources (RERs), as well as residential and industrial users. Cloud computing (CC) is required to process, access, store, and manage robust processing sources. Therefore, IoT and CC can be grouped together to form IoE platforms capable of empowering pervasive sensing. By storing and utilizing sensing data coherently, intelligent

observation and robust handling of the detection of data streams can be accomplished. With more smart appliances, latency and response time in CC increase, leading to deviations for several delay-sensitive devices and smart appliances [3]. In [4], authors developed realistic measures for assessing the physical structure of the cloud-fog-edge layers based on the reliability approach. They used the fault tree and Markov chain model to provide quantitative solutions. Ref. [5] suggested fog computing (FC) to address such problems. CC is dragged to the edge or corner of the network by fog or edge computing. By preprocessing the data, delay restrictions can be achieved, when meeting the scalability, interoperability, consistency, and enhanced system connectivity. Awareness of position, minimal delay, geographical distribution, a large number of appliances, mobility, real-time devices, and heterogeneity are some of the best attributes of FC [6]. The smart device system faces several issues, such as delayed demands, source-restricted devices, and network bandwidth. As the appliances have been linked to the internet, some safety issues arise, such as keeping source-restricted appliances safe, ensuring security, updating smart appliances, evaluating the safety state of large distributed systems accurately, and responding to vulnerabilities without any insurmountable problems which cannot be met by CC. Ref. [7] depicted the FC as a networking structure dispensing computing, storage, control, and system management near the appliances of consumers. The purpose of the study is to present a feasible framework for SG by integrating two new technologies: cloud and fog computing (CFC). Based on the proposed idea, a CFC-driven SG scheme was discussed in the paper, along with load-balancing strategies for handling SG consumer requirements.

SG structure is observable when two-way communication network systems are combined with electrical grids [8]. SGs are capable of providing electricity to consumers gradually and efficiently with the help of information and communication technology. According to SG technology, there are three main parts within the system, namely distribution, transmission, and production [9]. Obsolete power plants make up the production side. In addition, electric power must be distributed to consumers by the transmission feeders. In order to reduce environmental issues and the number of transmission losses caused by power plants, green energy or RERs have been introduced and gained popularity in recent years. Energy usage is monitored by the most important features of SG utilizing a variety of optimization strategies for the consumer [10,11]. Assuring the price, peak-to-average ratio, and consumer satisfaction are met by scheduling devices with various consumption profiles. Further, demand response management is emerging as an important empowerment technology for SGs.

In the literature, there are numerous studies that support the CC paradigm, such as ref. [12]. Ref. [13] compared CC with traditional approaches, recognizing several technical and nontechnical challenges and CC alternatives. The CC has been progressing quickly, but it has faced several problems, including long delays, poor reliability, mobility, awareness of position, and privacy issues [14,15]. Ref. [6] offered FC as a solution to the issues mentioned above. FC offers excellent solutions regarding latency, processing time, reliability, and so on. Technically, congestion and overburdening are caused by too many requests. This arbitrariness has its biggest drawback in task allocation. As a result of the random task allocation, there will be a few overloaded and a few underloaded processors [16]. As a result of load balancing, overloaded processes are directly transferred to underloaded ones. This problem is addressed by a number of experts, such as [17,18]. Ref. [19] examined the FC paradigm to optimize task planning with bee swarm algorithms aiming to optimize source usage.

CFC has been investigated as a way to aid in managing SG by many researchers. As a solution, a test case study was discussed for demand response optimization in the SG utilizing the CFC platform. As a result, CC's performance will be increased in the SG by using FC. Authors in [20] examined the SG information management system using the CC scheme, employing distributed information management for real-time information capturing, ubiquitous accessibility, and real-time data retrieval using parallel processing. In [21], a cloud-driven demand response framework was devised to enable quicker response

times in large-scale implementations. There were two kinds of demand response strategies investigated: cloud-driven demand response and dispersed demand response. Moreover, the convergence time was minimized and bandwidth was efficiently utilized. As it can be inferred from these explanations, none of the above works have investigated the role of CFC in SGs including the computational time needed to solve the problem. Therefore, this paper focuses on the use of computational intelligence (CI) methods for improving the efficiency of data transaction systems for multi-objective, multi-level optimization problems in CFC. There are two types of optimization problems, depending on the variables considered: combinatorial problems and continuous problems, respectively, for discrete variables and continuous variables. The paper discusses combinatorial optimization problems in SGs, presenting and analyzing the outcomes. Two novel hybrid algorithms including the gray wolf optimization (GWO) and improved differential evolution (IDE) are proposed which would decrease the computational burden effectively. The study's main contributions consist of:

- Developing new intelligent optimization algorithms for constructing fast and reliable computational algorithms needed in the CFC architectures including the hybrid versions of the gray wolf optimization (GWO) and improved differential evolution (IDE);
- Proposing a hybrid gray wolf differential evolution (HGWDE) optimization algorithm for SG energy management;
- Comparing the optimization approaches to help the cloud schedules for decreasing the response time and running time of tasks in CFCs.

The remainder of the study consists of below parts: the system scheme, problem definition, and suggested load-balancing method are discussed in Section 2. The suggested optimization approach is presented in Section 3. Section 4 provides a detailed description of the outcomes. The study is summarized and concluded in Section 5.

2. System Scheme

There are often several data centers located across the globe for cloud service providers [22]. The described system scheme of the CFC environments consists of several fog data centers and cloud data centers. An example of the SG model is shown in Figure 1, utilizing a geo-distributed CFC environment. Three layers are included in the proposed scheme: the end user, the fog layer, and the core cloud layer. The fog layer consists of a variety of software and hardware components to provide local analyses and monitoring. Users and the cloud are connected via this layer to each other. Moreover, the fog layer is responsible for managing requests from users from various parts of the globe. Data access and energy consumption are included in the requests. Therefore, the cloud load is reduced by the fog. The following part proposes architecture with 3 layers, consisting of a cloud, a user, and a fog. Data is shared between the 3 layers [23]. It is assumed that N buildings with several houses make up the user layer. All residences are equipped with an RER and an energy storage system (ESS) for meeting energy demands [24,25]. It is environmentally friendly and does not emit pollutants due to the fact that the energy is derived from natural resources. Furthermore, the ESS stores surplus produced power at low-production times for meeting domestic load demands [26]. A house's power consumption, production, and device planning are each received by the fog layer. Cloud services are used by the layer for operating their applications. The fog devices are connected to smart homes or households by smart meters. Each household communicates its energy shortage and surplus data via the CFC environments. The communication among the smart meters is performed through a local area network, a wide area network, or a metropolitan area network.

Cloud extensions such as fog act as a middleman between users and the cloud. All clusters are connected by the fog. Fog layers are used to temporarily store information, immediately prior to moving to the cloud for permanent storage. Fog provides similar characteristics as the cloud but can be more accessible to users. Hence, low-delay services are therefore beneficial to consumers. Virtual machines (VMs) in the fog process requests

from customers. A user can request power from microgrids by submitting a request to the fog.

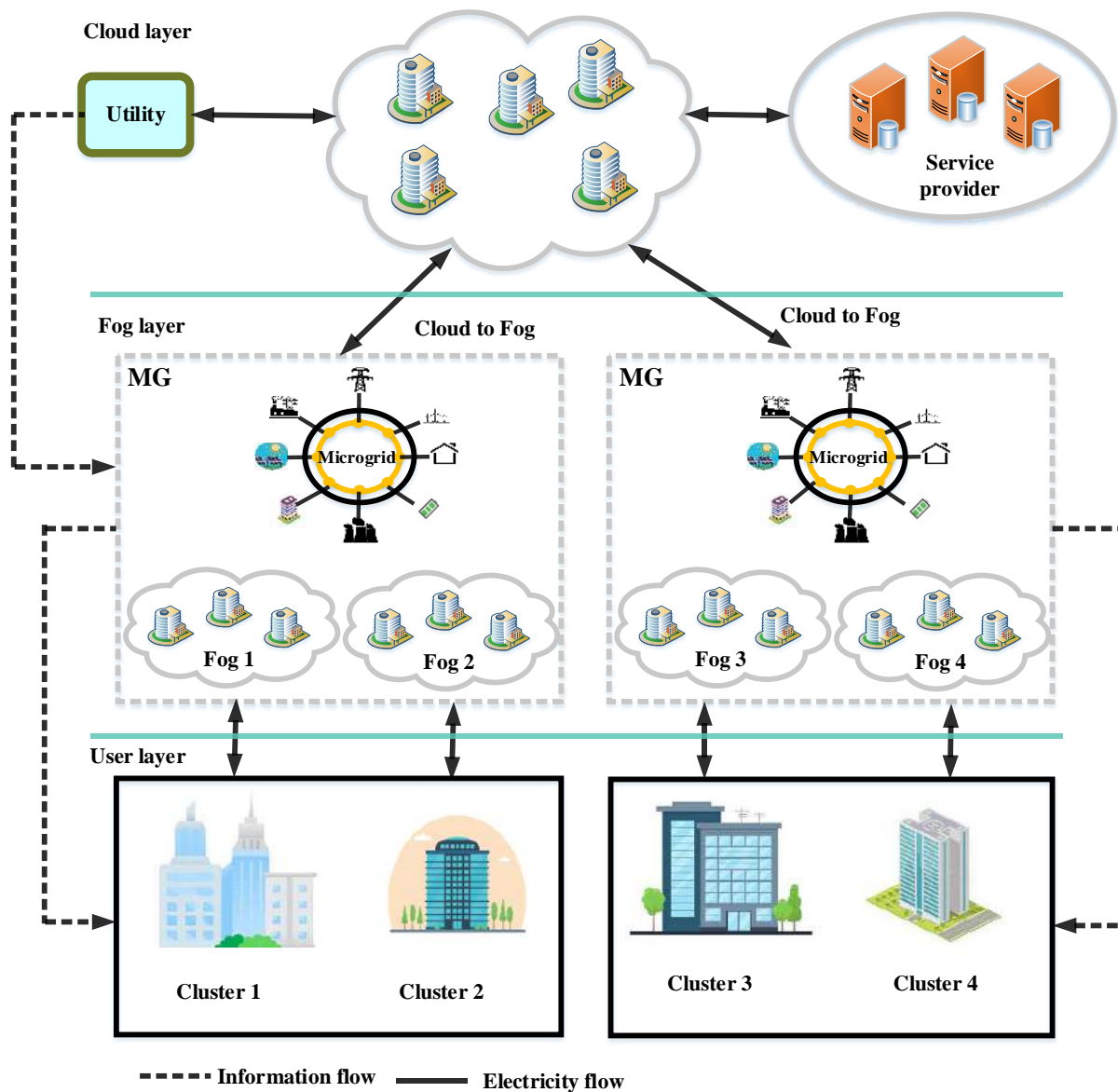


Figure 1. A representation of the suggested system scheme.

The second layer, called the fog layer, regulates delay and manages network sources effectively. As the fog layer has been located near the clients, it physically takes place at their local place. If the communication and physical distances are similar, the fog node will be near the user.

Another focus of this article is to propose an intelligent and fast cyber-layer for smart grids to record, store, analyze, and manage the big data all around the smart grid. In this way, first, the structure of the cloud-fog-user layer is depicted and described in Figure 1. Second, the appropriate model and algorithms required to control the proposed layer are described. Although these three layers look similar at first glance, they represent different computing resources as different layers of SG. In fact, each layer builds on the capabilities of the previous layer. In Figure 1, the lower level is the user layer, also called the edge layer, and is in charge of edge computing. In other words, this layer will handle some processing of sensor data away from the cloud layer (or central control of the SG). Therefore, the user layer pushes the computations to happen on the edge of the network. The input to this

layer can be named as the active and reactive power demand of the system loads, the voltage level of the system buses, the network topology, the renewable energy sources output power, the distributed generators capacity, price and limitations, the market price, etc. It will try not to let data transfer from the end users to the cloud for analysis of action. Therefore, the communication of big data toward the upper layers reduces and thus a simpler chain of communication with less potential for failures would be provided. In the second layer, the fog layer exists which extends cloud computing to a lower layer. This layer tries to push intelligence down to the end users and simulate an analyzing system close to the sensors in SG. Therefore, the fog layer is the first layer in the digital world of the smart grid. On the top of all layers, the cloud layer exists which has the capability of global storage of data, analyzing and launching final commands for the operation of the smart grid. The output of the upper cloud layer would be the power flow outputs including the voltage level of the buses, the power flow in the feeders, the optimal output power of the units, the renewable energy sources, and the operating point of the loads. These fogs have been controlled by internet service providers. To this end, various fogs are present in the fog layer. A fog device has been connected to all smart buildings in this way. A fog device consists of virtualized hardware (H/W). Through virtualization, VM monitors and manages multiple VMs on one computer. Virtual machines (VMs) manage multiple operating systems (OSs) on one H/W platform (VMM). VMMs or hypervisors are interfaces among guest OSs and VMs. VMs, or guest OSs, are central processing units that run multiple programs. User connections to the cloud are made through the fog.

The final layer includes the core cloud. In the cloud layer, information is processed and administered on demand via remote servers. There is an inextricable connection between CFCs. Consumer information is temporarily stored in fogs prior to being sent to the cloud to be permanently stored. A computing application's computational load profile plays a crucial role in the implementation of CC. Overburdened servers occur whenever there are many applications running on a single platform. According to Figure 2, consumers request numerous visits to the service providers. VMs utilize load balancers to maintain load balance and optimize source usage. Numerous load-balancing methods are used by CC to ensure effective control of computing load profiles. Computing load profiles are the same as power load profiles in SGs. Therefore, if the SGs are integrated with the CFC-driven environments, afterward, tasks associated with the SGs must also be efficiently managed in terms of the computing load profile. The paper uses 4 heuristic solutions to solve the load-balancing issue. In the case of the system scheme being applied to each part of the world, all areas have various numbers of structures and fogs.

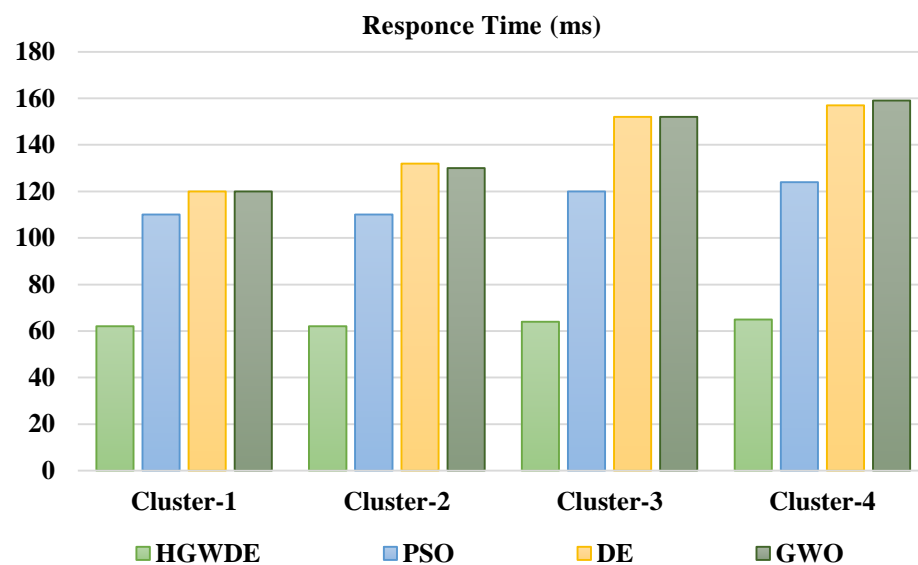


Figure 2. Average response time.

As can be seen from the cloud-fog-edge layer, the main purpose is to reduce the communication traffic and push the high computational burden down to the local areas. In this way, each layer should be equipped with appropriate fast and intelligent algorithms. The first task for any optimizer is to have a clear objective function. In Equations (1)–(10), the objective functions including the response time, the latency time, and the price of VMs are modeled. Later, new optimization evolving methods are developed to solve the optimization problem at the appropriate level. Through such an intelligent structure, the power delivery services would enhance greatly in the smart grid.

2.1. Problem Definition

There are 3 layers in the offered system framework. A representation of the offered structure is depicted in Figure 1. At the top, the cloud layer stands. The 2nd and sole intermediate layer would be the fog layer. The 3rd and last layers belong to the consumer. Hence, consumers' requirements are met by the layers communicating among themselves. The suggested structure consists of 6 distinct parts. Users send requests to the fog via the SG for computing and other measuring tasks. As a result of utilizing resources effectively, the fog meets consumer needs. The task group T is expressed as follows:

$$T = T_1, T_2, \dots, T_m \quad (1)$$

Fogs can have the following number of VMs:

$$VM = \sum_{i=1}^v (VM_i) \quad (2)$$

wherein v shows the number of VMs. The objective function consists of minimizing response time and processing, expressed in the following way:

$$K_{minimizw} = \sum_{j=1}^m \sum_{i=1}^n (RT * P_{ij} * Delay) \quad (3)$$

wherein RT shows the response time as in Equation (6) and $Delay$ shows the delay time. The fitness function can be determined in the following way:

$$Fitness = Max[EXC_{vm_1}(F_1), \dots, EXC_{vm_n}(F_m)] \quad (4)$$

In which, $EXC_{vm_1}(F_1)$ shows the working time for performing the group of tasks on fog F_1 on vm_1 . F_1 shows the group of clusters of consumers, meaning, $F_1 = [C_1, C_2, \dots, C_x]$, in which x is the count of consumers on fog F_1 . In addition, n shows the count of VMs and m shows the count of fogs.

2.1.1. Processing Time

VM capacity and task length processing time are determined in the following manner:

$$PT = \sum_{i=1}^N \sum_{j=1}^M (P_{ij} * A_i) \quad (5)$$

2.1.2. Response Time

The response time has been determined by the difference between the times that the task began to execute and when the customer transmitted the request:

$$RT = DelayTime + FinishTime - ArrivalTime \quad (6)$$

2.1.3. Price

The cost of CFC is also an essential parameter. In addition to the *MG* cost, *VM* cost and data transfer (*DT*) cost are also two factors that the cost depends on. The below formulas are used to calculate the total cost.

$$Cost_{Total} = Cost_{DT} + Cost_{VM} + Cost_{MG} \quad (7)$$

As can be seen in Equation (7), the total cost is computed based on the summation prices of *DT*, *VM*, and *MG*, where $Cost_{DT}$ represents the cost of *DT*, $Cost_{VM}$ shows the cost of *VM*, and $Cost_{MG}$ defines the cost of *MG*. *VM* cost and *DT* cost are obtained by Equations (8) and (9), respectively, as follows:

$$Cost_{VM} = \sum_{i=1}^N (VM_{FinalTime} - VM_{InitialTime}) * u \quad (8)$$

$$Cost_{DT} = \frac{T_{Total}}{Data_{Used} * \beta} \quad (9)$$

where u shows a fixed factor and β shows a per *GB* transferring price. $VM_{FinalTime}$ and $VM_{InitialTime}$ are the final time and start time of *VM*, respectively. $Data_{Used}$ shows the transmitted data which is used to execute. *VM*'s overall time is shown in Equation (10):

$$Total_{Time} = Finish_{Time} - Start_{Time} \quad (10)$$

where $Total_{Time}$ defines the overall time of *VM*. $Start_{Time}$ and $Finish_{Time}$ are the start time and finish time of *VM* execution.

3. Suggested Model

This paper proposes HGWDE as an optimization method that combines two meta-heuristics: IDE and GWO for optimizing energy usage. The three methods have been employed for comparing the efficiency of the suggested model to that of current models.

3.1. Improved Differential Evolution (IDE)

IDE shows a new version of DE, offered by Storn in 1995 for the first time. In the algorithm, the primary population has been determined at random. IDE involves four major stages: population production, mutation, crossover, and choice. Equation (11) generates the population at random.

$$X_{i,j} = l_j + (rand \times (u_j - l_j)) \quad (11)$$

The goal is to produce a mutant vector by generating three vectors x_{r1} , x_{r2} , x_{r3} . The target vector belongs to the 1st vector. Equation (12) gives the mutant vector in the following way:

$$V_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) \quad (12)$$

In which, F shows a scaling factor. Following the creation of the mutant vector, 3 test vectors have been produced. Next, the optimal test vector has been chosen for comparing it to the target vector, thereby populating the production with the optimal vector. Equations (13)–(15) can be used to create the first 3 test vectors.

$$u_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & \text{if } randb(j) \leq 0.3 \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (13)$$

$$u_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & \text{if } randb(j) \leq 0.6 \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (14)$$

$$u_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & \text{if } randb(j) \leq 0.9 \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (15)$$

Equations (16) and (17) can be used to create the 4th and 5th test vectors.

$$u_{j,i,G+1} = randb(j) \cdot x_{j,i,G} \quad (16)$$

$$u_{j,i,G+1} = randb(j) \cdot v_{j,i,G} + (1 - randb(j)) \cdot x_{j,i,G} \quad (17)$$

Max. *iter* shows the maximal number of iterations; *POP* shows the whole population, i.e., all potential solutions to home energy management (HEM); and *h* shows the whole time intervals. CR shows the crossover factor, usually 0.3, 0.6, and 0.9. *u*, *μ*, and *x* show the mutant vector, test vector, and target vector. Table 1 shows IDE with regard to HEM.

Table 1. IDE mapping on HEM.

IDE variables	Population	Number of dimensions	Gradient of problem
HEM variables	Feasible solution	Number of appliances	Scheduling
Amounts	50	17	Vary

3.2. Gray Wolf Optimization (GWO)

Wolf hunting behavior and leadership hierarchies are the inspirations for GWO's meta-heuristic optimization method. Leadership can be divided into 4 levels: alpha α , beta β , delta δ , and gamma γ . As the group's smartest leader, alpha guides the others on how to hunt. Alpha is followed by delta and beta under the hierarchical structure, and gamma defines at the bottom. Thus, gamma is ineligible for leadership positions. In HEM, alpha would be considered the fittest member to minimize costs. Equation (18) creates a random population at the start:

$$X(i, j) = rand(POP, D) \quad (18)$$

In which, *POP* is the entire gray wolf population, and *D* represents all counts of devices. In every search agent, the objective function (the distance from the target) is determined using *A* and *C*.

3.2.1. Encircling Target

Target is encircled by gray wolves prior to the hunt. According to ref. [27], the below formulas can be used to formulate gray wolves' encircling behavior.

$$X(t+1) = X_p(t) - A \times D \quad (19)$$

$$D = |C \times X_p(t) - X(t)| \quad (20)$$

Here X_p shows the location of the target, whereas X shows the location of the wolf for the t th iteration, and can be determined via Equation (19). Moreover, Equations (21) and (22) can be used to calculate the vectors *A* and *C*:

$$\vec{A} = 2\vec{a} \times \vec{r}_1 - \vec{a} \quad (21)$$

$$\vec{C} = 2 \times \vec{r}_2 \quad (22)$$

In which, \vec{r}_1 and \vec{r}_2 show randomly selected vectors in the range [0, 1]. Several iterations reduce *a*'s value from 2 to 0. *C* varies at random within a range of 0 and 2, indicating the weight for target attractiveness [28].

3.2.2. Hunting

Alpha is the primary guide to the hunt, with beta and delta serving as secondary factors. The alpha leads them because the alpha knows the location of the target most well.

Gamma updates its location based on the optimal solution determined by the 1st three members. Equation (23) updates the location of wolves.

$$\vec{X}_{t+1} = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3} \quad (23)$$

Equations (24)–(26) can be used to determine \vec{x}_1 , \vec{x}_2 , and \vec{x}_3 .

$$\vec{x}_1 = \vec{x}_\alpha - A_1 \times \left(\vec{d}_\alpha \right) \quad (24)$$

$$\vec{x}_2 = \vec{x}_\beta - A_2 \times \left(\vec{d}_\beta \right) \quad (25)$$

$$\vec{x}_3 = \vec{x}_\delta - A_3 \times \left(\vec{d}_\delta \right) \quad (26)$$

Here \vec{x}_α , \vec{x}_β , and \vec{x}_δ show the optimal solutions achieved in the t th iteration; Equation (21) can be used to determine \vec{A}_1 , \vec{A}_2 , and \vec{A}_3 , and Equations (27)–(29) can be used to determine \vec{D}_α , \vec{D}_β , and \vec{D}_δ :

$$\vec{D}_\alpha = \vec{C}_1 \times \vec{x}_\alpha - \vec{x} \quad (27)$$

$$\vec{D}_\beta = \vec{C}_2 \times \vec{x}_\beta - \vec{x} \quad (28)$$

$$\vec{D}_\delta = \vec{C}_3 \times \vec{x}_\delta - \vec{x} \quad (29)$$

Equation (22) can be used to determine C_1 , C_2 and C_3 . In the final step, variable a is graded from two to zero in all iterations, controlling the balance between exploitation and exploration based on Equation (30).

$$a = 2 - t \frac{2}{Max.iter} \quad (30)$$

The objective function is shown in Equation (31) and can be determined by multiplying the power rating by the state of every device.

$$Fitness = \rho_D \times Xv_D(h) \quad (31)$$

Max. iter shows the maximal iterations, POP shows the entire population, D shows the count of devices, and fitness would be the objective function. α shows the optimal solution or participant in the hunting process, whereas β and δ would be the 2nd and 3rd optimum solutions.

The fitness function has been compared to the fitness of α , β , and δ for evaluating the optimal leader. Equations (27)–(29) can be used to update their last locations.

3.3. Hybrid Gray Wolf Differential Evolution (HGWDE)

Please note that the new hybrid algorithm called HGWDE is a direct contribution of this work which is specifically designed for the optimal operation of the SG. The proposed algorithm benefits from the hybridization of the GWO and DE with a new formulation to limit the computational time needed for big data analysis and transactions within the CFC architecture. A detailed discussion of the suggested model has been provided here. IDE generates an updated population through four stages: initialization, mutation, crossover, and choice. In order to update the population, the optimal test vector will be chosen from five vectors and then compared to the target vector. Since IDE takes into account the entire range of test vectors, it is efficient at selecting the optimal trial vector. Encircling targets, hunting, and updating wolf locations make up GWO's main stages. The leader belongs to

α . Each search agent updates its location based on α . GWO does not compare α to β or δ . Moreover, β and δ may be nearer to the target than α . Crossover from IDE must be used to compare each search agent. GWO is used to update the location of search agents after the optimal search agent has been selected. The HGWDE was chosen because it combines the most advantageous characteristics of IDE and GWO.

Initialization, encircling the target, selecting the optimal search agent, and updating the location are the stages of HGWDE. To this end, the population of wolves $X_i (i = 1, 2, \dots, n)$ is first generated randomly. Then, three randomly selected vectors are used to create a mutant vector v . α , β , and δ are initially set up as three vectors. Equation (31) calculates the fitness of v and α , β , and δ . The crossover will be carried out according to the following formulas:

$$\alpha_{new} = \begin{cases} v_j & \text{if fitness of } v_j \leq \alpha \\ \alpha & \text{Otherwise} \end{cases} \quad (32)$$

$$\beta_{new} = \begin{cases} v_j & \text{if fitness of } v_j \leq \beta \\ \beta & \text{Otherwise} \end{cases} \quad (33)$$

$$\delta_{new} = \begin{cases} v_j & \text{if fitness of } v_j \leq \delta \\ \delta & \text{Otherwise} \end{cases} \quad (34)$$

The location of search agents generally changes following a selection process. Table 2 illustrates the mapping of HGWDE parameters with HEM.

Table 2. HGWDE mapping on HEM.

Variable	HEM			
	Feasible Solution	Devices	Minimum (Price)	State of the Devices
Amount	50	17	vary	1 or 0
Variable	HGWDE			
	Population	Wolfs in all packs	Minimal distance form target	State of the leader
Amount	50	17	vary	1 or 0

In the beginning, the needed parameters are initialized. Stage 2 generates the population at random. Iterations are adjusted to max once the population has been generated. IDE is used to compare the mutant vector's fitness to α , β , and δ . GWO updates the location of search agents until the stopping criteria are met.

4. Simulation Results and Explanation

This process is likewise implemented as part of Cloud-Analyst, a Cloud-Sim toolkit extension. A graphical user interface is available for Cloud-Analyst as well. In Cloud-Sim, VMs are simulated and deployed with various hardware features. Regarding the simulation results analysis and modeling, the main focus of this article is to assess the role of meta-heuristic algorithms to help the three-layer CFC architecture improve the process of analyzing data, storing data, and managing them in a proper time with the least possible latency. Therefore, the modeling process first comes around the task division among the layers as explained before which happens in the Cloud-Sim toolkit. Any further modeling would come around the optimization algorithms and how they can analyze big data, solve the problem considering the technical limits, and get to the optimal solution with the least computational time and highest accuracy. It is clear that modeling can expand to many more details up to communication channels, the communication traffic layers, etc., which are out of the scope of this paper. A comprehensive analysis and comparison is made among the different algorithms which are explained in the rest of this study.

4.1. Response Time Validation

Figure 2 depicts the response time of user clusters with PSO, DE, GWO, and HGWDE algorithms. As can be seen from Figure 2, PSO, DE, and GWO have much longer response times than HGWDE. Optimal response times were achieved for each user cluster at the same time. The fog is reduced by HGWDE's efficient allocation of resources. Using simulated annealing, HGWDE finds the optimal possible solution for all jobs and plans requests efficiently. As soon as a job arrives, the load balancer determines the VM's memory, utilization, energy requirement, and speed. In addition, according to the variables, jobs are allocated to the VM with the highest priority, which keeps the procedure moving quickly. The outcomes show that it is considerably better than PSO, DE, and GWO in terms of load balancing. Based on the outcomes, it can be concluded that odds and GWO have similar results. PSO outperformed the odds and GWO [29,30]. Table 3 illustrates the summary of the response times of each method. The HGWDE algorithm outperforms various algorithms.

Table 3. Overview of response time for each algorithm.

Algorithms	HGWDE	PSO	DE	GWO
Max. (ms)	87.3	584.7	581.4	577.1
Min. (ms)	38.7	41.8	42.6	42.6
Average (ms)	63	116	144.1	143.6

A PSO reaction takes 116.01 ms, a DE reaction takes 144.10 ms, a GWO reaction takes 143.60 ms, and an HGWDE reaction takes 63.02 ms, according to Figure 3.

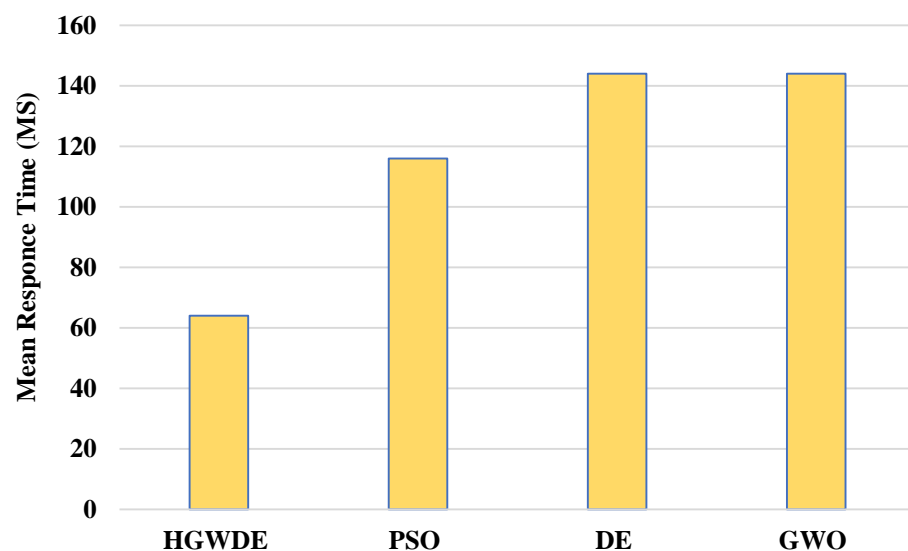


Figure 3. Average response time for clusters.

4.2. Processing Time Validation

Figure 4 depicts the processing time of user clusters with the PSO, DE, GWO, and HGWDE algorithms. It can be seen from Figure 4 that the processing time for the PSO, DE, and GWO exceeds that for the HGWDE. Optimal processing times were achieved for each user cluster at the same time.

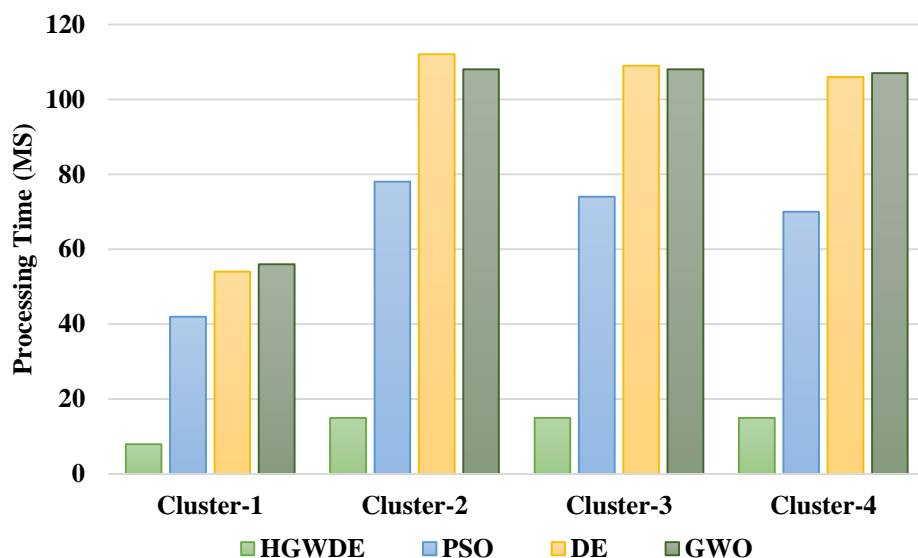


Figure 4. Comparing processing time of various algorithms.

In HGWDE, many agents, known as particles, work together to optimize the fog’s load. A deterministic and stochastic component helps the particles find the optimal solutions. It also performs better because it has fewer coefficients to tune than other optimization techniques. With each iteration, a particle gets nearer to the final ideal solution.

According to Figure 5 and Table 4, the average processing time for the PSO is 66.33 ms, for DE is 94.59 ms, for GWO is 93.95 ms, and for HGWDE is 13.39 ms.

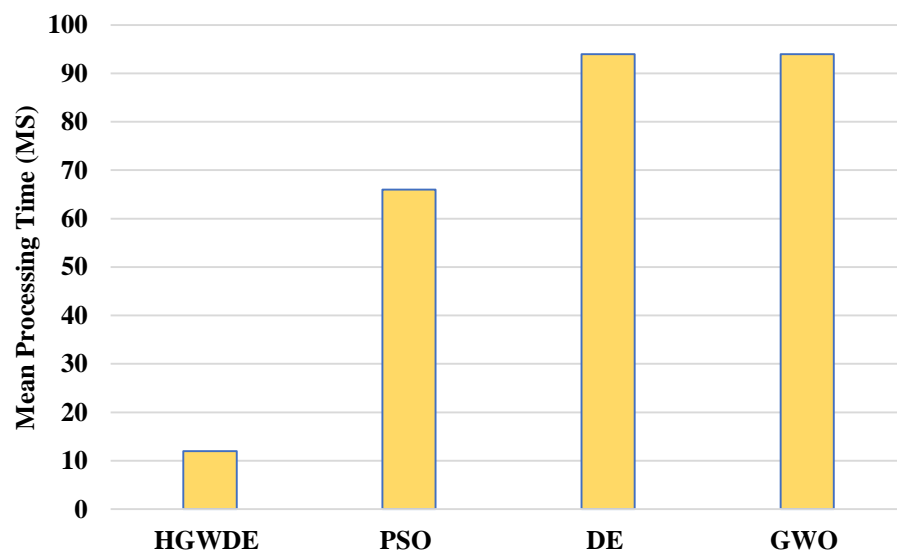


Figure 5. Average processing time of fogs.

Table 4. Overview of processing time for each algorithm.

Algorithms	HGWDE	PSO	DE	GWO
Max. (ms)	26.1	531.6	530.5	530.3
Min. (ms)	0.2	0.3	0.6	1.1
Average (ms)	13.4	66.3	94.6	93.9

4.3. Price and Costs Validation

The VMC is shown in Figure 6. The VMC for the user clusters with the PSO, DE, GWO, and HGWDE algorithms is shown in Figure 7. Figure 7 shows that the price of the PSO,

DE, and HGWDE exceeds that of the GWO. Optimal prices were achieved for each user cluster at the same time.

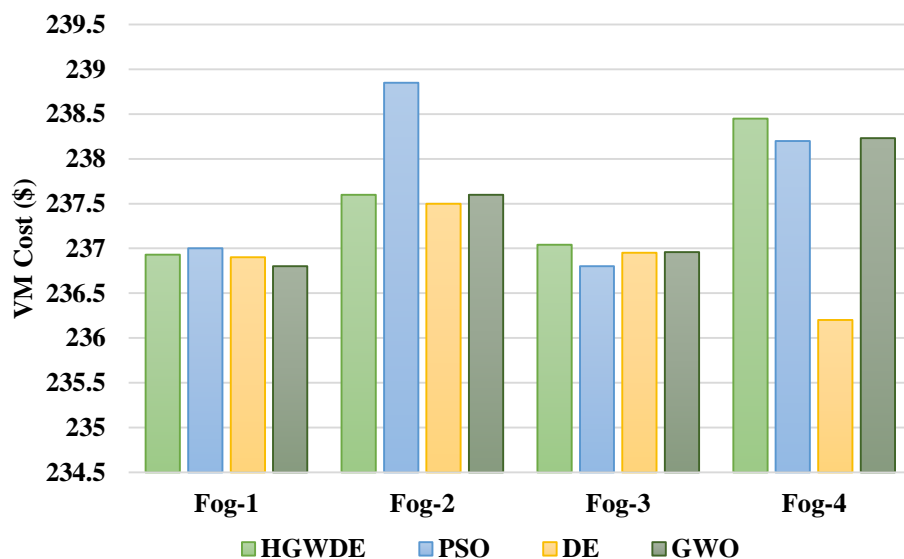


Figure 6. Average VMC for fogs.

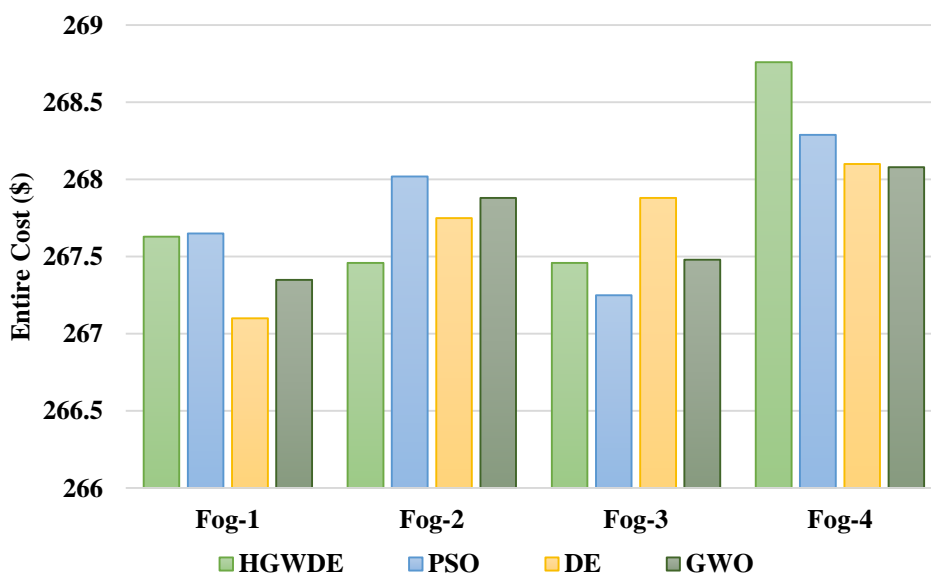


Figure 7. Average overall price for fogs.

Based on the outcomes, GWO provides a more cost-effective load-balancing algorithm in comparison to PSO, DE, and HGWDE. GWO and DE methods have similar outcomes, according to the results. With the DE approach, the PSO and GWO performed better. Table 5 provides a summary of the prices associated with the earlier described algorithms.

Table 5. Overview of price for each algorithm.

Algorithms	HGWDE	PSO	DE	GWO
Overall price (USD)	1071.3	1071.2	1070.9	1070.8
Information transfer price (USD)	121.3	121.3	121.3	121.3
VM price (USD)	950	949.9	949.6	949.5

According to Figure 8, the average VMC for PSO is 949.9\$, for DE is 949.5\$, for GWO is 949.5\$, and for HGWDE is 950\$. According to Figure 9, the average price for PSO is 1071.9\$, for DE is 1070.9\$, for GWO is 1070.8\$, and for HGWDE is 1071.3\$.

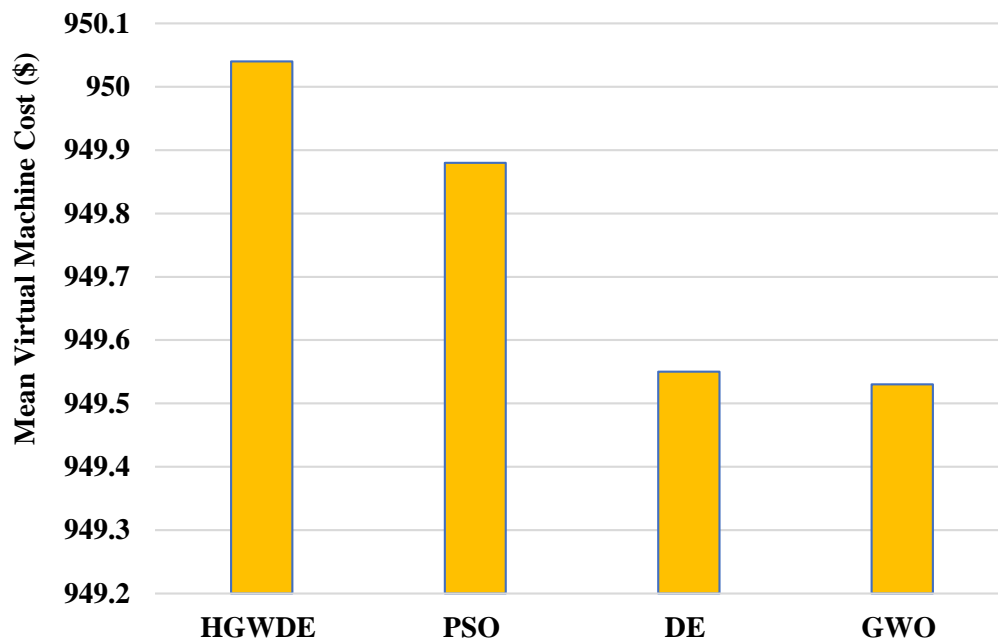


Figure 8. Average VMC.

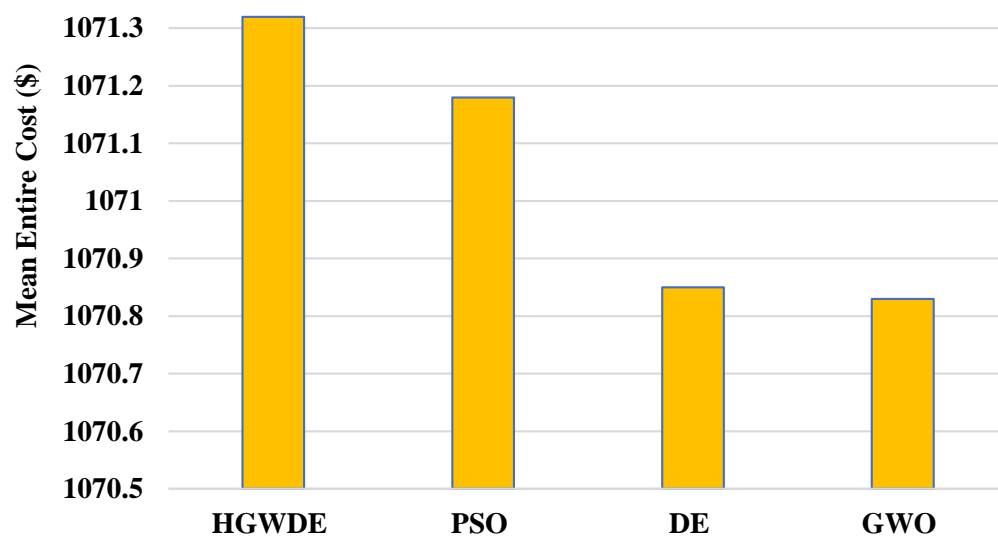


Figure 9. Average price.

Simulations indicate that HGWDE performs better than the PSO, GWO, and DE algorithms. As a result of combining PSO's superior features with simulated annealing, HGWDE exhibits superior efficiency. HGWDE provides the optimal global and GWO provides local solutions. Response time, processing time, running time, and price of the GWO were slightly higher because of its slow convergence. Local optima may prevent GWO from determining the global optimal solution. As a way to meet GWO limitations in the future, ABC fitness could be added to the algorithm, resulting in faster reaction times, processing times, costs, and execution times as a result of its higher convergence rate.

5. Conclusions

This paper presents an intelligent and optimal structure for a CFC-driven environment integrated with SG. There are three layers in the scheme: a cloud, a fog, and a consumer. The cloud layer consists of cloud servers, the fog layer consists of fog servers and VMs, and the end-user layer contains buildings, which will cover a number of home consumptions. A novel scheme using improved DE and GWO algorithms, called HGWDE, was proposed to reduce the latency and enhance the model efficiency. In HGWDE, the inertia weight is used to regulate search space sizing so that the optimal solution can be found. A comparison is made among HGWDE, PSO, DE, and GWO. There is a slight advantage for GWO in terms of price performance, but the difference is not significant. The simulation results clearly show the high performance and capabilities of the proposed model for managing smart grids. Moreover, it was seen that the successful performance of an SG relies on big data transaction management and analysis which is made possible only through a fast, precise, and reliable architecture. The proposed CFC is a necessity for the optimal and successful performance of the SG considering the big data transactions happening within the system. The simulation results clearly advocate the necessity of smart and intelligent optimization algorithms embedded within the CFC, which is proposed by HGWDE in this paper. Not only the accuracy of the proposed hybrid algorithm is validated through several comparisons made for the optimal scheduling of the SG, but also very little computational effort is needed to show its fitting role for CFC architecture. The authors would address the effect of the proposed cyber-layer based on metrics and measures in future works.

Author Contributions: Conceptualization, F.A., A.A., M.A.M. and M.M.; Methodology, M.A.M.; Software, M.A.M.; Formal analysis, M.A.M.; Investigation, F.A., A.A., M.A.M. and M.M.; Data curation, M.A.M.; Writing—original draft, F.A., A.A., M.A.M. and M.M.; Writing—review & editing, A.A. and M.A.M.; Supervision, F.A. and M.A.M.; Funding acquisition, F.A. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the support provided by King Abdullah City for Atomic and Renewable Energy (K.A.CARE) under the K.A.CARE-King Abdulaziz University Collaboration Program. The authors are also thankful to the Deanship of Scientific Research, King Abdulaziz University for providing financial support vide grant number (RG-49-135-42). Furthermore, the authors would like to thank the Estonian Centre of Excellence in Zero Energy and Resource Efficient Smart Buildings and Districts, ZEBE, for grant TK146, funded by the European Regional Development Fund to support this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the support provided by King Abdullah City for Atomic and Renewable Energy (K.A.CARE) under the K.A.CARE-King Abdulaziz University Collaboration Program. The authors are also thankful to the Deanship of Scientific Research, King Abdulaziz University for providing financial support vide grant number (RG-49-135-42). Furthermore, the authors would like to thank the Estonian Centre of Excellence in Zero Energy and Resource Efficient Smart Buildings and Districts, ZEBE, for grant TK146, funded by the European Regional Development Fund to support this research.

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

Nomenclature

T	Task group	X_{ij}	Population
m	Count of fogs	X_p	Location of the target
v	Number of virtual machines	\vec{r}_1 and \vec{r}_2	Randomly selected vectors between [0, 1]
n	Count of VMs	α	Best search agent
m	Count of fogs	β	2nd best optimal solution
x	Count of consumers on fog F_1	δ	3rd best optimal solution
u	Fixed factor	$\vec{x}_\alpha, \vec{x}_\beta$ and \vec{x}_δ	Optimal solutions achieved in the t^{th} iteration
β	Per GB transferring price	D_α	Position of the α search agent
P_{ij}	Length of task	D_β	Position of the β search agent
A_i	Task specified for every request	D_δ	Position of the δ search agent
Arrival _{Time}	The time when the task began	F	Scaling factor
Finish _{Time}	The time when the task execution is finished	v	Mutant vectors
Delay _{Time}	Delay time from customer transmitting the request till the task began to execute	Max. iter	Maximal number of iterations
Cost _{DT}	Cost of DT	POP	Whole population
Cost _{VM}	Cost of VM	h	Whole time intervals
Cost _{MG}	Cost of MG	CR	Crossover factor
VM _{final time}	The final time of VM	u	Mutant vector
VM _{InitialTime}	Start time of VM	μ	Test vector
Data _{Used}	Transmitted data which is used to execute	x	Target vector
Total _{Time}	The overall time of VM	D	All counts of devices
P_{ij}	Length of task	A and C	Fitness coefficients
K_{minimizw}	Minimizing response time and processing	X	Location of the wolf for the t^{th} iteration
EXC _{vm₁} (F_1)	Working time of performing the group of tasks on fog F_1 on VM ₁	a	Graded from two to zero in all iterations
F_1	Group of clusters of consumers	PT	Processing time
Start _{Time}	The start time of VM execution	RT	Time duration from when a request of uses is sent to fog and when a response has been obtained
Finish _{Time}	The finish time of VM execution	VM	Virtual machines
l_j and u_j	Lower and upper bounds		

References

- Askari, M.; Dehghani, M.; Razmjoui, P.; GhasemiGarpachi, M.; Tahmasebi, D.; Ghasemi, S. A novel stochastic thermo-solar model for water demand supply using point estimate method. *IET Renew. Power Gener.* **2022**, *16*, 3559–3572. [\[CrossRef\]](#)
- Wang, B.; Ma, H.; Wang, F.; Dampage, U.; Al-Dhaifallah, M.; Ali, Z.M.; Mohamed, M.A. An IoT-Enabled Stochastic Operation Management Framework for Smart Grids. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1025–1034. [\[CrossRef\]](#)
- Chen, J.; Alnowibet, K.; Annuk, A.; Mohamed, M.A. An effective distributed approach based machine learning for energy negotiation in networked microgrids. *Energy Strategy Rev.* **2021**, *38*, 100760. [\[CrossRef\]](#)
- Nguyen, T.A.; Min, D.; Choi, E.; Lee, J.-W. Dependability and security quantification of an internet of medical things infrastructure based on cloud-fog-edge continuum for healthcare monitoring using hierarchical models. *IEEE Internet Things J.* **2021**, *8*, 15704–15748. [\[CrossRef\]](#)
- Balasubramanian, S.; Meyyappan, T. Game theory based offload and migration-enabled smart gateway for cloud of things in fog computing. In *Computing in Engineering and Technology*; Iyer, B., Deshpande, P.S., Sharma, S.C., Shiukar, U., Eds.; Springer: Singapore, 2020; pp. 253–266.
- Lai, K.-L.; Chen, J.I.Z.; Zong, J.I. Development of smart cities with fog computing and internet of things. *J. Ubiquitous Comput. Commun. Technol. UCCT* **2021**, *3*, 52–60. [\[CrossRef\]](#)
- Sabireen, H.; Neelananarayanan, V. A review on fog computing: Architecture, fog with IoT, algorithms and research challenges. *Ict Express* **2021**, *7*, 162–176.
- Kazmi, S.; Javaid, N.; Mughal, M.J.; Akbar, M.; Ahmed, S.H.; Alrajeh, N. Towards optimization of metaheuristic algorithms for IoT enabled smart homes targeting balanced demand and supply of energy. *IEEE Access* **2017**, *7*, 24267–24281. [\[CrossRef\]](#)
- Khan, S.; Paul, D.; Momtahan, P.; Aloqaily, M. Artificial intelligence framework for smart city microgrids: State of the art, challenges, and opportunities. In Proceedings of the 2018 3rd International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, Spain, 23–26 April 2018.
- Jamil, M.; Mittal, S. Hourly load shifting approach for demand side management in smart grid using grasshopper optimisation algorithm. *IET Gener. Transm. Distrib.* **2020**, *14*, 808–815. [\[CrossRef\]](#)

11. Jokar, H.; Bahmani-Firouzi, B.; Simab, M. Bilevel model for security-constrained and reliability transmission and distribution substation energy management considering large-scale energy storage and demand side management. *Energy Rep.* **2022**, *8*, 2617–2629. [[CrossRef](#)]
12. Wei, Y.; Blake, M.B. Service-oriented computing and cloud computing: Challenges and opportunities. *IEEE Internet Comput.* **2010**, *14*, 72–75. [[CrossRef](#)]
13. Shetty, J.P.; Panda, R. An overview of cloud computing in SMEs. *J. Glob. Entrep. Res.* **2021**, *11*, 175–188. [[CrossRef](#)]
14. Mohammadi, M.; Kavousi-Fard, A.; Dehghani, M.; Karimi, M.; Loia, V.; Alhelou, H.H.; Siano, P. Reinforcing Data Integrity in Renewable Hybrid AC-DC Microgrids from Social-Economic Perspectives. *ACM Trans. Sens. Netw.* **2022**. [[CrossRef](#)]
15. Zhang, Q.; Fu, S.; Jia, N.; Xu, M. A verifiable and dynamic multi-keyword ranked search scheme over encrypted cloud data with accuracy improvement. In *Security and Privacy in Communication Networks; International Conference on Security and Privacy in Communication Systems*; Beyah, R., Chang, B., Li, Y., Zhu, S., Eds.; Springer: Cham, Switzerland, 2018.
16. Zhang, W.-Z.; Elgendy, I.A.; Hammad, M.; Iliyasa, A.M.; Du, X.; Guizani, M.; El-Latif, A.A.A. Secure and optimized load balancing for multitier IoT and edge-cloud computing systems. *IEEE Internet Things J.* **2020**, *8*, 8119–8132. [[CrossRef](#)]
17. Afzal, S.; Kavitha, G. Load balancing in cloud computing—A hierarchical taxonomical classification. *J. Cloud Comput.* **2019**, *8*, 22. [[CrossRef](#)]
18. Lilhore; Kumar, U.; Sarita, S.; Kalpna, G.; Devendra, P. An efficient load balancing method by using machine learning-based VM distribution and dynamic resource mapping. *J. Comput. Theor. Nanosci.* **2020**, *17*, 2545–2551. [[CrossRef](#)]
19. Arri, H.S.; Singh, R. Energy optimization-based optimal trade-off scheme for job scheduling in fog computing. In *Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 17–19 March 2021.
20. Albogamy, F.R.; Ashfaq, Y.; Hafeez, G.; Murawwat, S.; Khan, S.; Ali, F.; Khan, F.A.; Rehman, K. Optimal Demand-Side Management Using Flat Pricing Scheme in Smart Grid. *Processes* **2022**, *10*, 1214. [[CrossRef](#)]
21. Saleem, M.U.; Usman, M.R.; Politis, C. Design, deployment and performance evaluation of an IoT based smart energy management system for demand side management in smart grid. *IEEE Access* **2022**, *10*, 15261–15278. [[CrossRef](#)]
22. Akram, J.; Tahir, A.; Munawar, H.S.; Akram, A.; Kouzani, A.Z.; Mahmud, M.A.P. Cloud-and Fog-Integrated Smart Grid Model for Efficient Resource Utilisation. *Sensors* **2021**, *21*, 7846. [[CrossRef](#)]
23. Shojaeiyan, S.; Niknam, T.; Nafar, M. A novel bio-inspired stochastic framework to solve energy management problem in hybrid AC-DC microgrids with uncertainty. *Int. J. Bio-Inspir. Comput.* **2021**, *18*, 165–175. [[CrossRef](#)]
24. Mohamed, M.A.; Almalaq, A.; Abdullah, H.M.; Alnowibet, K.A.; Alrasheedi, A.F.; Zaindin, M.S.A. A distributed stochastic energy management framework based-fuzzy-PDMM for smart grids considering wind park and energy storage systems. *IEEE Access* **2021**, *9*, 46674–46685. [[CrossRef](#)]
25. Chen, W.; Liu, B.; Nazir, M.S.; Abdalla, A.N.; Mohamed, M.A.; Ding, Z.; Bhutta, M.S.; Gul, M. An energy storage assessment: Using frequency modulation approach to capture optimal coordination. *Sustainability* **2022**, *14*, 8510. [[CrossRef](#)]
26. Ramos, F.; Pinheiro, A.; Nascimento, R.; Junior, W.d.A.S.; Mohamed, M.A.; Annuk, A.; Marinho, M.H.N. Development of Operation Strategy for Battery Energy Storage System into Hybrid AC Microgrids. *Sustainability* **2022**, *14*, 13765. [[CrossRef](#)]
27. Zhang, X.; Lin, Q.; Mao, W.; Liu, S.; Dou, Z.; Liu, G. Hybrid Particle Swarm and Grey Wolf Optimizer and its application to clustering optimization. *Appl. Soft Comput.* **2021**, *101*, 107061. [[CrossRef](#)]
28. Chen, Z.; Jin, T.; Zheng, X.; Liu, Y.; Zhuang, Z.; Mohamed, M.A. An innovative method-based CEEMDAN-IGWO-GRU hybrid algorithm for short-term load forecasting. *Electr. Eng.* **2022**, *104*, 3137–3156. [[CrossRef](#)]
29. Mohamed, M.A.; Chabok, H.; Awwad, E.M.; El-Sherbeeney, A.M.; Elmeligy, M.A.; Ali, Z.M. Stochastic and distributed scheduling of shipboard power systems using M θ FOA-ADMM. *Energy* **2020**, *206*, 118041. [[CrossRef](#)]
30. Rizwan, M.; Waseem, M.; Liaqat, R.; Sajjad, I.A.; Dampage, U.; Salmen, S.H.; Al Obaid, S.; Mohamed, M.A.; Annuk, A. SPSO Based Optimal Integration of DGs in Local Distribution Systems under Extreme Load Growth for Smart Cities. *Electronics* **2021**, *10*, 2542. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.