❒ 477

Hybrid algorithm for cloud-fog system based load balancing in smart grids

Afaf Saoud, Abdelmadjid Recioui
Laboratory of Signals and Systems, Institute of Electrical and Electronic Engineering, University M'hamed Bougara of Boumerdes,
Boumerdes, Algeria

Article Info

ABSTRACT

Energy management is among the key components of smart metering. Its role is to balance energy consumption and distribution. Smart devices integration results in a huge data exchange between different parts of the smart grid causing a delay in the response and processing time. To overcome this latency issue, the cloud computing has been proposed. However, cloud computing does not perform well when there are large distances from the cloud to the consumers. Fog computing solves this issue. In this paper, a cloud-fog computing system is presented to achieve an accurate load balancing. The hybridization of whale optimization algorithm with bat algorithm (WOA-BAT) is proposed for load balancing. The model performance is compared to state of art load balancing techniques as throttled, round robin, whale and particle swarm optimization algorithms in terms of processing and the response time. The results reveal that the proposed WOA-BAT has better results in terms of response time than the three algorithms with 4.3% improvement compared to RR and TH. It also outperforms all the algorithms in terms of processing time by at least 22.3%.

Corresponding Author:

Afaf Saoud
Laboratory of Signals and Systems, Institute of Electrical and Electronic Engineering
University M'hamed Bougara of Boumerdes, Boulevard de l'indépendance, 35000, Boumerdes, Algeria
Email: a.saoud@univ-boumerdes.dz

1. INTRODUCTION

Smart grids (SGs) have revolutionized the management of generation, transmission and distribution in the electricity grid. Smart grids incorporate the advanced two-way communication technology along with computing systems to improve the electrical energy distribution, consumption and management. They have proved to ameliorate the performance of the grid with regards to safety, cost, reliability and efficiency [1], [2]. With the continuous development and growth of the societies, cities and communities; the energy demand has increased leading to more requests from the consumers' side that has become a real challenging problem to governments [3]. Dealing with requests affects the processing performance of the system and may cause latency in the overall process. Energy management is among the key components of smart metering system that deals with the latter issue. Its main goal is to ensure the right balance between the consumed energy and the supplied one. Nevertheless, the involvement of smart devices leads to enormous data traffic among the different parts of the smart grid causing a delay in the response and processing time [4].

To overcome this latency issue, the cloud computing has been presented to process these demand requests [5]. The optimization of the energy supply and consumption is done by the cloud based energy management system and the IoT devices [6]. An ensemble of data centers or a large data center positioned at different regions may constitute the cloud that treats and executes the user demand requests based on the cluster computing resources. The cloud can be hosted by several sectors such as enterprises, governments and

service providers [7]–[9]. Cloud computing techniques are important in the smart systems as they connect different parts of the system via internet and ease the use of resources, storing data and processing tasks [10]. Cloud computing affords multiple services in private, public and hybrid manners. It incorporates services such as the platform as a service (PaaS) that comprises software and hardware devices allowing the development of applications, the infrastructure as a service (IaaS) composed of software, hardware and different IT infrastructure connected via the internet, and the software as a service (SaaS) that affords applications with availability to users through the internet [1]. The cloud data center is a group of physical machines (PM) containing several virtual machines (VMs) [11]. Smart grids require an efficient energy management realized by the combination of computing infrastructure and communication network. The energy management engenders the arrival of a great number of daily requests from the users which should be processed and computed via cloud computing [12]. However, the extension of the cities requires large number of IoT devices being at the origin of latency and maximizes the response time while making the communication privacy harder because of the increased number of requests and the distance spacing between the consumers and the cloud platform.

In order to alleviate these problems, fog computing has emerged [5], [13], [14]. The introduced fog with cloud computing manages the large number of the smart devices by proceeding and storing data in the cloud [15]. The fog-cloud computing hierarchy is composed of three layers where the fog is the layer that links between the cloud and the users' layers by assigning resources nearer to the clients to minimize the response time. Fog computing has several advantages such as enhancing security, minimizing computation cost, reducing latency and high speed processing [16], [17]. In real-time applications, Fog computing [18]–[20] performance improvement is important as the demand in energy exceeds its capacity, it engenders high latency and data loss. One solution to these issues is the microgrid (MG). These MGs are located near the consumers and satisfy their demands [10]. When the consumers transmit requests to fog through smart meters, the nearest MG located by fog ensure the power supply to the consumers. Managing and handling the client requests is a crucial role in Fog computing. Therefore, task scheduling, load balancing, has become the difficult step to study and needs an efficient technique in cloud fog computing [21]. An effective task scheduler is known by its adaptation to the environment changing and the task types [22]. An optimized task scheduling can ameliorate the performance of the fog cloud computing.

In real-time applications, data can be connected with the local datacenters close to the users [22]. With the development of the electric grid and introducing the smart grid concept, a centralized cloud platform is introduced to handle the technology use increase [23]. However, the cloud suffers from the long response time and storage. Fog computing fixes these issues [22]. Despite the several advantages of fog, fog computing suffers from the load balancing problem due to the great number of applicants [24]. Load balancing in fog and cloud computing has become the hot spot of research of multiple studies. James and Verma [25] thave worked on "iterative turns" algorithm that assign tasks on VMs iteratively. Mondal *et al.* [26] proposed the random hill climbing method for an improved load balancing in cloud datacenters by optimizing the resources. Sran and Kaur [27] achieved the task scheduling based on min-min method. Another technique was used in [28] for load balancing using honey bee colony algorithm to get a quick response to requests and an efficient management of VMs. [29] presented a service broker policy and compared its performance with throttled and round robin load balancing algorithms. The authors dealt with load balancing using shortest job first method was proposed [30]. The particle swarm optimization [31] was used in [32], [33] to maximize the efficiency of load balancing that enhance the fog performance.

Many studies have been done regarding fog and cloud load balancing based on meta-heuristic algorithms [34]. PSO performance was enhanced in multiple studies by combining it with other algorithms to overcome its drawbacks in global solution. The authors presented PSO non-dominated genetic algorithm (PSONSGA) was presented to fix the problem of the multi-objective optimization at the fog stage in [35]. PT and RT of the cloud and fog computing edges were minimized in [5] by proposing the PSO-simulated annealing (PSO-SA) for load balancing and a new service broker policy (NSBP). Another algorithm that has been used widely in load balancing is the ant colony optimization (ACO). The ACO was proposed to improve the performance of the cloud computing in [36]. Guo [37] has improved the task scheduling for cloud computing using the improved ACO, multi-objective ant colony algorithm (MO-ACO), to allocate tasks on resources optimally to decrease the PT and RT. Another hybrid algorithm based on ACO was introduced in [38]. The hybrid ACO-Cuckoo algorithm has reduced the task execution time [38]. Another algorithm was proposed that is Dragonfly algorithm for load balance and task scheduling in datacenters of cloud computing in [39]. Goyal *et al.* in [40] have enhanced the load balancing and resource scheduling based the meta-heuristic method, whale optimization algorithm (WOA). WOA has high performance to solve multiple optimization tasks. However, it has some convergence problem [41]-[45]. For this reason, the BAT algorithm may be implied with WOA to enhance exploration capability of WOA [42].

In this paper, a cloud-fog computing system is introduced for a distributed buildings spread over different regions. The load balancing of the VMs in Fogs is improved by introducing the hybrid WOA-BAT optimization algorithm. The performance of the suggested model is compared to the performances of other models as throttled (TH), round robin (RR), whale optimization (WOA) and particle swarm (PSO) methods. The evaluation metrics are the processing time (PT) and response time (RT).

The remaining of the paper is presented as: section 2 is the introduction of the system model and problem formulation. Section 3 presents the suggested methodology. The results of the work are presented and discussed in section 4. The paper is concluded in section 5 by providing the impact of this work and eventual future perspective.

2. SYSTEM MODEL AND PROBLEM FORMULATION

The model of the system proposed in this work is illustrated in the Figure 1. It comprises three layers from top to bottom: a centralized cloud layer, a distributed fog layer and the consumer layer. The costumers' layer includes groups of buildings. Each home in the building has a smart meter. In this work, three regions are considered. In each region, there is a fog that communicates with buildings giving three fogs in total in the model. There are three buildings in the cluster of region 1, two buildings in the cluster of region 2 and one building in region 3. The clusters of customers transmit energy demand requests to fog hourly via smart meters. The communication between fog layer and customer layer is achieved based on the wireless communication as Zigbee, Z-Wave and Wi-Fi. The received requests by fogs are assigned to VMs using load balancing technique based on the availability and the capacity to handle the requests. Fogs keep an information log about the customer from its metadata. Metadata is the request information as the location ID, load consumption, load demand, cost, source of energy, time, number and power ratings of appliance, etc. The request information differs depending on the client services from the fogs. Fog connects with MG for the energy supply to the house in demand. In case that MG has insufficient energy, fog connects with the cloud to satisfy the demand of this region. The cloud on its turn connects with the SG service provider. SG locates the nearest MG to the area in need for energy supply.
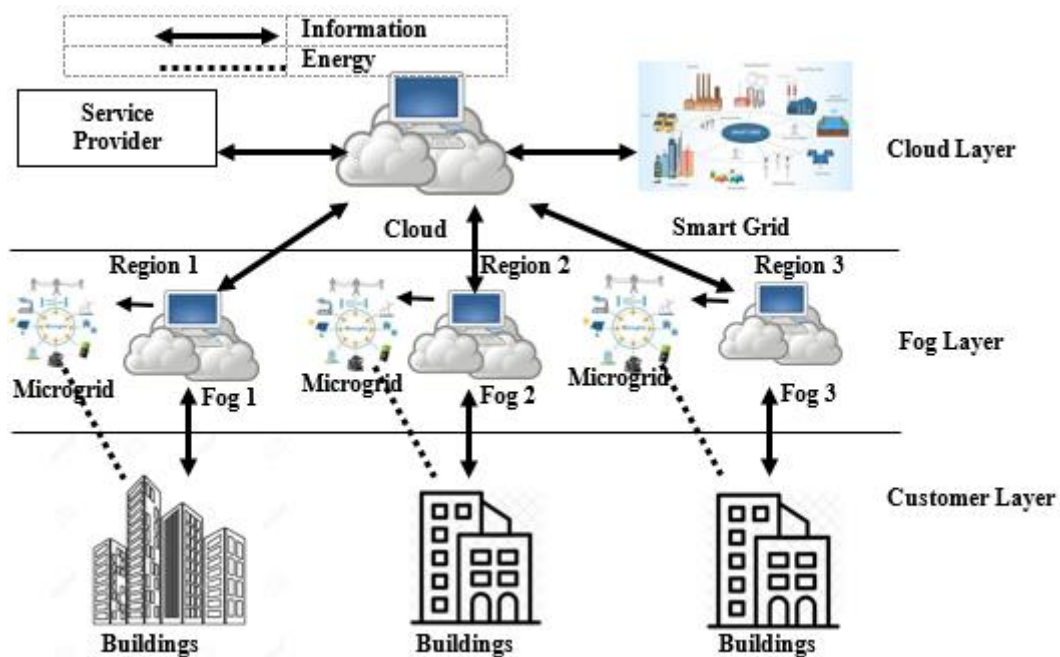

Figure 1. Proposed model of the system

Assume we have in each region M fogs. Every fog comprises m VMs where VM={VM_1, VM_2, VM_3, . . . , VM_m} is the set of VMs. VMs process the tasks n T that are requested by the clients and assigned by the load balancing algorithm. T={t_1, t_2, t_3, . . . , t_n} is the set of tasks. The VMs have the same capacity to process several tasks. They work in parallel. Assigning VMs is based on the evaluation the performance metrics: response time, processing time, cost of VM and requests per hour. The formulation of these

parameters is done linear programming [46], [47]. In this paper, we are concerned with the enhancement of PT and RT. The binary representation in (1) checks if the task t is assigned to VM. When no task is assigned to VM it will be 0. 1 shows if task is assigned to VM.

$$t_{ij} = \begin{cases} 1; if\ VM\ is\ assigned \\ 0; \qquad\qquad else \end{cases} \tag{1}$$

Among the objectives of this work is minimizing the PT_t. PT_t is the total processing time of VM, where tij is the ith requests assigned to j^{th} VM in specific time period. PT_{ij} is the processing time of the request i on the VMj. In (2) represents the objective function of PT and its constraints.

$$Minimize\ PT_t = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(PT_{ij} * t_{ij}\right) \tag{2}$$

$$subject\ to:\ \sum_{i=1}^{n} t_{ij} = 1, j = 1,2,\dots,n$$

$$t_{ij} = 0\ or\ 1$$

The other objective is minimizing the total response time of VM. (3) is used to calculate RT_t. CT_x represents the completion time of request. The *Makespan* term describes the maximum completion time required for a request. The makespan of i task on VMj is denoted by CTi,j.

$$Minimize\ RT = \frac{\sum VM_S(CT_x)}{makespan * Num\ of\ VM_S} \tag{3}$$

$$MakeSpani = Max(CTi,j)$$

3. THE PROPOSED ALGORITHM

3.1. Whale optimization algorithm

The whale optimization algorithm has been proposed by Mirijalili and Lewis for simulating the hunting conduct of humpback whales. It is based on two mechanisms of attack: firstly, they chase the prey with the best search agent or random (exploitation stage) and secondly they simulate the bubble net hunting approach (exploration phase). The targets of the Humpback whales are the small fish near to the surface. Hence, they make a winding shaped way by swimming around the aim alongside and inside a thin circle, forming by that different blebs along a '9' or circle shaped ways on the whole as illustrated in Figure 2. This behavior is named the bubble net feeding technique [48]. The mathematical representation [49] of each stage is demonstrated in the coming subsections.
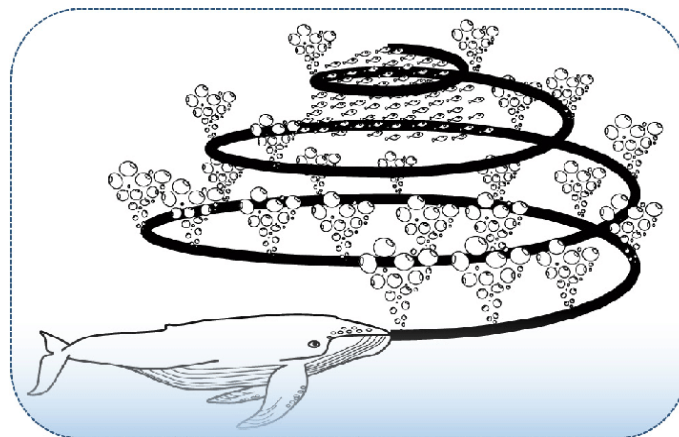

Figure 2. Spiral shape bubble net

3.1.1. Bubble net attacking technique

The bubble net behavior is mathematically modeled by two stages: the exploitation stage and the exploration stage.

a. Encircling the prey

After locating the prey, the humpback whales surround them. At the beginning, the WOA supposes that the actual leading nominee solution is close to the optimum or the target prey due to the non identification of the optimal design' position in the search plan. After that, the rest of the agents will try to update their positions to the best search agents. The following equations represent this behavior:

$$\vec{X}(t + 1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D} \tag{4}$$

$$\vec{D} = \left|\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{5}$$

$\vec{X}(t + 1)$ is the whale's present location, $\overrightarrow{X^*}(t)$ is the whale's previous best position. $\vec{X}(t)$ is the position vector and $\vec{D}$ represents the distance vector between prey and whale. The coefficient vectors C and A are computed with:

$$\vec{A} = 2.\vec{a}.\vec{r} + \vec{a} \tag{6}$$

$$\vec{C} = 2.\vec{r} \tag{7}$$

In (6) reduces the value of $\vec{a}$ Resulting in the reduction of $\vec{A}$. In the course of optimization iterations, a is decreased from 2 to 0 and the random vector is in the range [0, 1]. The updated position of any agent will be designated in the interval between the original location of agent and the location of the actual best agent by assigning values for $\vec{A}$ in (−1, 1) randomly.

b. Spiral position updating

After computing the distance between the prey (X*, Y*) and the whale (X, Y), a spiral equation is produced between the position of the prey and the whale to copy the displacement of humpback whales in helix-shaped manner as:

$$\vec{X}(t + 1) = e^{bk}.\text{Cos}(2\pi k).\overrightarrow{D^*} + \overrightarrow{X^*}(t) \tag{8}$$

$$\overrightarrow{D^*} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{9}$$

K is random number in the interval [−1, 1] and b is a constant that identifies the shape of logarithmic spiral. The selection between the spiral model and the shrinking encircling mechanism has 50% of chance as follows where p is in the range (0, 1):

$$\vec{X}(t + 1) = \begin{cases} \overrightarrow{X^*} - \vec{A}.\vec{D} \ if \ p < 0.5 \\ e^{bk}.\text{Cos}(2\pi k).\overrightarrow{D^*} + \overrightarrow{X^*}(t) \ if \ p \geq 0.5 \end{cases} \tag{10}$$

3.1.2. Search for prey

In the exploration stage, whales try to find their prey based on the location of each other using a random search giving $\vec{A}$ random values less or greater than 1. In the exploration stage, the random selection of the search agent reorganize the position of the search agent instead the best search agent in the exploitation stage. This approach overcomes the local optimal problem and helps to get the global search as demonstrated in (11) and (12):

$$\vec{X}(t + 1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \tag{11}$$

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_{rand}} - \vec{X}\right| \tag{12}$$

Giving the random position vector (whale) X_{rand} is set from the present population.

3.2. Bat algorithm

Bat algorithm has been developed by Yang [50]. He simulated the bat echolocation abilities concept for solving multiple optimization issues. The emission of loud sound wave and receiving the echo coming back from the preys help bats to find the location of their preys [51]. They fly in the search space randomly with Vi speed and change their locations Xi at a constant f_{min}, loudness A0 and several wavelengths β. The values of the parameters are updated as:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{13}$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_*)f_i \tag{14}$$

$$x_i^{t+1} = x_i^t + v_i^t \tag{15}$$

f_i Is the frequency of waves, vi is the speed of bats, β is a random vector in the interval [0, 1], x_i Is the location of bats, and x* is the global best solution. Usually, f_{min} Takes 0 and f_{max} Is 100. First, each single bat a frequency from the interval (f_{min}, f_{max}). There is compatibility between the frequency of the search agent and the velocity. The location of the solution is updated based on the new velocity. When the prey is found, the loudness rate is reduced and the pulse emission increases.

3.3. Hybrid WOA-BAT algorithm

WOA has high performance in resolving different optimization tasks. However, it has some drawbacks as the slow speed of the convergence because of searching for the global optimum [20]. Consequently, the exploration of WOA is enhanced by the BAT algorithm. The hybridization of these two algorithms depends on two methods: i) embedding the BAT method inside of WOA search stage and ii) when the location of every search agent is changed, the condition method is implied. In case the new location is better than the previous location, then the previous location is updated. The WOA-BAT flowchart is illustrated in Figure 3.
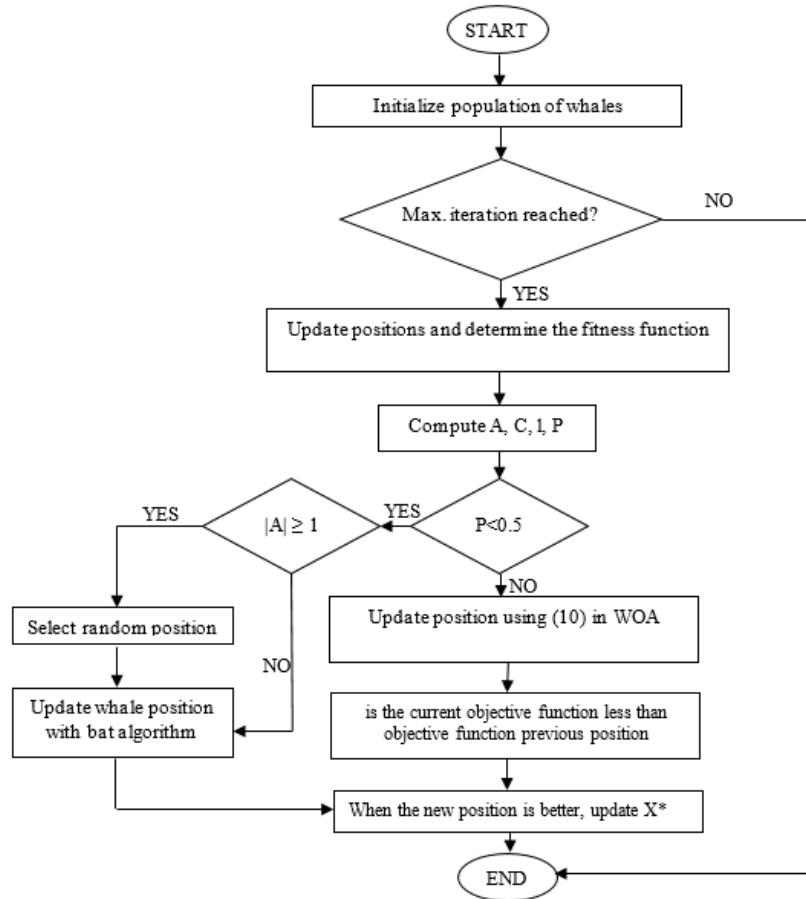

Figure 3. Flowchart of WOA-BAT algorithm

3.4. Proposed load balancing mechanism

The great number of requests arrived to the cloud and fog is executed by resources and the efficient use of resources improves the performance of the computing system. In this paper, the requests are processed in fog according to VMs performance. An optimized execution of the whole load of requests is done by

allocating the requests to VMs efficiently. In order to get this result, load balancing algorithms are introduced. Load balancing is demonstrated in Figure 4. The process is from top to bottom. The consumers send requests to the fog data center controller. This last submits requests to load balancer that runs the load balancing algorithm in order to assign requests to virtual machines. VMs are created on physical machines and controlled by VM manager. The PT and RT with our load balancing algorithm WOA-BAT are evaluated and compared to those of other load balancing algorithms throttled, round robin, whale and particle swarm optimization algorithm.
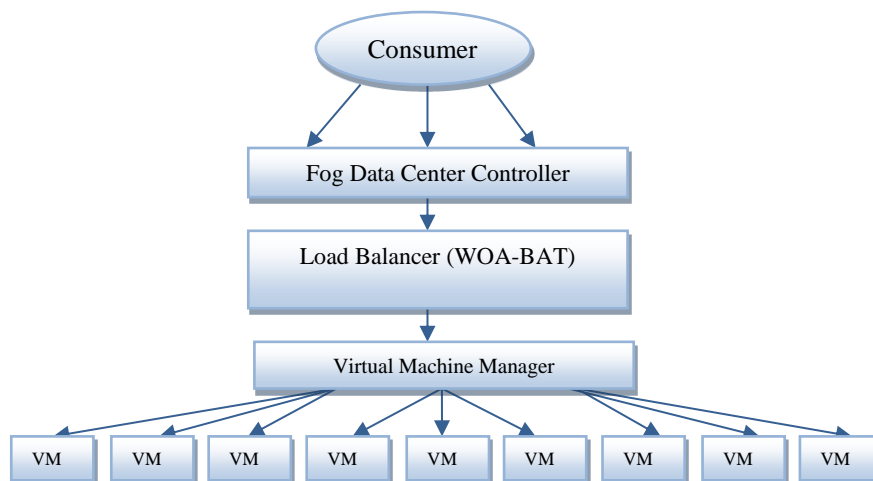
Figure 4. Proposed load balancing model

4. RESULTS AND DISCUSSIONS

In this work, we have chosen three different regions for the customers' base. Each region has one fog center; in total we have three fogs. The fogs are connected to one cloud data center. The simulations were performed on Java platform in Netbeans and cloud analyst tool [52] for one day (24h). The parameters settings on cloud analyst tool are illustrated in Table 1. The buildings in the simulation are referred to user base (UB). The number of houses in buildings of the user bases ranges from 80 to 120. Number of VMs per Fog is 15. The simulation of this system was performed using five algorithms: RR, TH, PSO, WOA and our proposed model WOA-BAT. The evaluation metrics for this work are RT and PT of data centers. The resource allocation policy was performed based on the closest data center service broker.

Table 1. User base settings

	Region	Request per hour	Data size per req	Peak hour Start	Peak hour end	Avg peak Users	Avg off Peak user
UB1	0	100	250	6	11	1000	100
UB2	0	300	250	6	11	1000	100
UB3	0	200	250	7	12	1000	100
UB4	1	200	250	7	12	1000	100
UB5	1	250	250	8	2	1000	100
UB6	2	250	250	8	2	1000	100

4.1. Simulation results; response time

The simulation results of the average response time RT of the six buildings for one day are illustrated in the Table 2 and Figure 5. The buildings send different number of energy demand requests to fog of their region due to their power needs. PSO has the highest RT with 58.50 ms, 58.53 ms, 58.47 ms, 58.62, 57.72 ms and 57.71 ms for UB1, UB2, UB3, UB4, UB5 and UB6, respectively. RR and TH have better results than PSO. TH outperforms RR with a small percentage around 0.1%. WOA has better results than the three algorithms with 4.3% of improvementin RT as compared to RR and TH. The proposed model WOA-BAT does better than all the algorithms including WOA by 3.3% giving rise to the lowest RT for all buildings 51.16 ms, 51.18 ms, 51.15 ms, 51.20 ms 51.90 ms and 51.89 for UB1, UB2, UB3, UB4, UB5 and UB6, respectively. Another issue that should be pointed out here is the number of requests processed by the

VMs in Fogs and their impact on RT. The UB6 has the least RT since it is the only building connected to Fog3 with 250 requests per hour.

Table 2. The average response time of different load balancing algorithm

	RR	TH	PSO	WOA	WOA-BAT
UB1	54.14	54.09	58.50	52.36	51.16
UB2	54.16	54.12	58.53	52.38	51.18
UB3	54.11	54.07	58.47	52.35	51.15
UB4	54.25	54.20	58.62	52.40	51.20
UB5	53.48	53.47	57.72	52.09	51.90
UB6	53.41	53.40	57.71	52.08	51.89
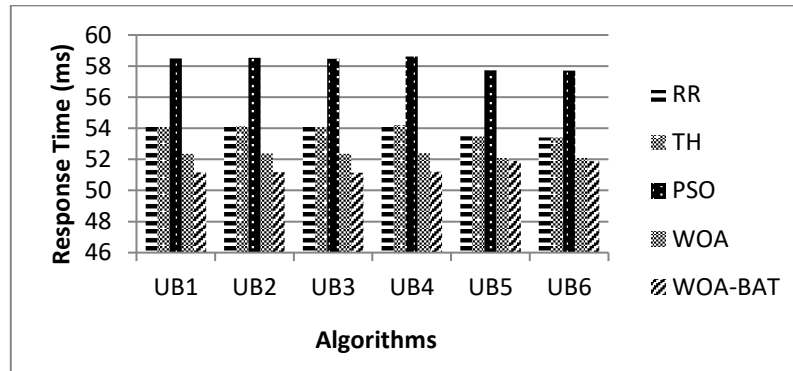
Figure 5. The average response time of the six buildings based on RR, TH, PSO, WOA and WOA-BAT

4.2. Simulation results; processing time

Figure 6 and Table 3 illustrate the results of the average processing time of the three fogs based on RR, TH, PSO, WOA and WOA-BAT. Fog1 has the highest average processing time and Fog3 has the least average processing time and these results are obtained with all the algorithms due to the highest number of requests arrived to Fog1 as compared to the number of requests arrived to Fog3. PSO has the highest PT with the three fogs with 6.42 ms, 6.04 ms and 5.04 for Fog1, Fog2 and Fog3, resectively. RR and TH outperfom PSO and have approximate perfomance with better results obtained by TH with around 0.9% of enhancement as compared to RR. WOA outperforms the three algorithms giving 3.87, 3.65 and 2.86 for for Fog1, Fog2 and Fog3, resectively. The proposed load balancing WOA-BAT has the lowest average PT. It outperforms all the algorithms by 33%, 32.2%, 53.2% and 22.3% for RR, TH, PSO and WOA, respectively.
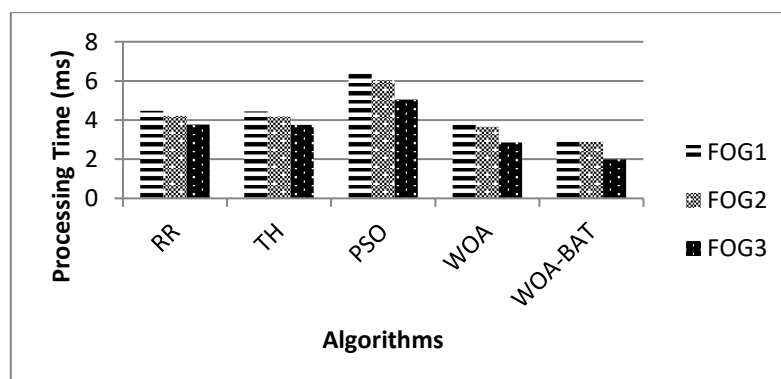

Figure 6. The average excution time of different load balancing algorithm

Table 3. The average excution time of different load balancing algorithm

	RR	TH	PSO	WOA	WOA-BAT
FOG1	4.48	4.44	6.42	3.87	3.01
FOG2	4.21	4.18	6.04	3.65	2.89
FOG3	3.77	3.76	5.04	2.86	1.98

5. CONCLUSION

In this paper, a system model for cloud-fog to enhance the load balancing in smart grids has been proposed. Fog computing turns out to improve the latency and the response time. We proposed WOA-BAT algorithm for load balancing and its performance has been contrastedwith respect to four algorithms: RR, TH, PSO and WOA. The comparison was based on quantitative metrics that are the processing time (PT) and the response time (RT) of the system for each algorithm. The results demonstrate that the proposed algorithm outperfoms the other algorithms. It turns out to provide the lowest RT and PT. For future contributions, the proposed algorithm may be used to schedule appliances and microgrids to improve the smart grid performance. Another use of this algorithm may be with a new service broker policy for better results.

REFERENCES

[1] A. Saoud and A. Recioui, "A review on Data communication in smart grids," *Algerian Journal of Signals and Systems*, vol. 2, no 3, pp. 162-179, sept. 2017, doi: 10.51485/ajss.v2i3.42.

[2] A. Recioui and H. Bentarzi, *Optimizing smart grid operation and control*, Pennsylvania, United States: IGI Global, 2021.

[3] Z. A. Khan *et al.*, "Energy Management in Smart Sectors Using Fog Based Environment and Meta-Heuristic Algorithms," in *IEEE Access*, vol. 7, pp. 157254-157267, 2019, doi: 10.1109/ACCESS.2019.2949863.

[4] A. Recioui and F. Z. Dekhandji, "Implementation of Load Control for Smart Metering in Smart Grids," In: *Optimizing and Measuring Smart Grid Operation and Control*, A. Recioui and H. Bentarzi, IGI Global, 2021, doi: 10.4018/978-1-7998-4027-5.ch006.

[5] R. Bukhsh, N. Javaid, Z. Khan, F. Ishmanov, M. Afzal and Z. Wadud, "Towards Fast Response, Reduced Processing and Balanced Load in Fog-Based Data-Driven Smart Grid," *Energies*, vol. 11, no. 12, pp. 3345, 2018, doi: 10.3390/en11123345.

[6] M. Hans and V. Jogi, "Peak load scheduling in smart grid using cloud computing," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1525-1530, 2019, doi: 10.11591/eei.v8i4.1687.

[7] R. K. Naha and M. Othman, "Cost-aware service brokering and performance sentient load balancing algorithms in the cloud," *Journal of Network and Computer Applications*, vol. 75, pp. 47–57, 2016, doi: 10.1016/j.jnca.2016.08.018.

[8] D. Bernstein, D. Vij and S. Diamond, "An Intercloud Cloud Computing Economy - Technology, Governance, and Market Blueprints," *2011 Annual SRII Global Conference*, 2011, pp. 293-299, doi: 10.1109/SRII.2011.40.

[9] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," *2009 Fourth International Conference on Internet and Web Applications and Services*, 2009, pp. 328-336, doi: 10.1109/ICIW.2009.55.

[10] M. Zahid, N. Javaid, K. Ansar, K. Hassan, M. K. Khan and M. Waqas, "Hill Climbing Load Balancing Algorithm on Fog Computing. In Proc.: *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, vol. 24, pp. pp 238-251, 2018, doi: 10.1007/978-3-030-02607-3_22.

[11] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang and Q. Zheng, "Power-Aware and Performance-Guaranteed Virtual Machine Placement in the Cloud," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1385-1400, 1 June 2018, doi: 10.1109/TPDS.2018.2794369.

[12] C. Stergiou, K.E. Psannis, B.-G. and Kim, B. Gupta, "Secure integration of IoT and Cloud Computing," *Future Generation Computer Systems,* vol. 78, part 3, pp. 964-975, January 2018, doi: 10.1016/j.future.2016.11.031.

[13] J. Kim and Y. Kim, "Benefits of cloud computing adoption for smart grid security from security perspective," *The Journal of Supercomputing, Springer*, vol. 72, no. 9, pp. 3522–3534, September 2016, doi: 10.1007/s11227-015-1547-0.

[14] M. A. Al Faruque and K. Vatanparvar, "Energy Management-as-a-Service Over Fog Computing Platform," in *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 161-169, April 2016, doi: 10.1109/JIOT.2015.2471260.

[15] F. Chang, J. Ren and R. Viswanathan, "Optimal Resource Allocation in Clouds," *2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 418-425, doi: 10.1109/CLOUD.2010.38.

[16] M. M. Hussain, M. S. Alam, and M. M. Beg, "Fog computing in iot aided smart grid transition-requirements, prospects, status quos and challenges," *2nd EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing*, 2018.

[17] C. Shi *et al.*, "Ultra-low latency cloud-fog computing for industrial Internet of Things," *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1-6, doi: 10.1109/WCNC.2018.8377192.

[18] Q. Fan and N. Ansari, "Towards Workload Balancing in Fog Computing Empowered IoT," in *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 253-262, 1 Jan.-March 2020, doi: 10.1109/TNSE.2018.2852762.

[19] S. Bitam, S. Zeadally and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm," *Enterprise Information Systems*, vol. 12, pp. 373-397, 2018, doi: 10.1080/17517575.2017.1304579.

[20] M. Adhikari and T. Amgoth, "Heuristic-based load-balancing algorithm for iaas cloud," *Future Generation Computer Systems,* vol. 81, pp. 156-165, doi: 10.1016/j.future.2017.10.035.

[21] Q. Huang and T. Huang, "An optimistic job scheduling strategy based on QoS for Cloud Computing," *2010 International Conference on Intelligent Computing and Integrated Systems*, 2010, pp. 673-675, doi: 10.1109/ICISS.2010.5655492.

[22] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," in *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854-864, Dec. 2016, doi: 10.1109/JIOT.2016.2584538.

[23] G. Ramadhan, T. W. Purboyo and R. Latuconsina, "Experimental model for load balancing in cloud computing using throttled algorithm," *International Journal of Applied Engineering Research*, vol. 13, no. 2, pp. 1139-1143, 2018.

[24] S. Bera, S. Misra and J. J. P. C. Rodrigues, "Cloud Computing Applications for Smart Grid: A Survey," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1477-1494, 1 May 2015, doi: 10.1109/TPDS.2014.2321378.

[25] J. James and B. Verma, "Efficient VM Load Balancing Algorithm for a Cloud Computing Environment," *International Journal on Computer Science and Engineering*, vol. 4, no. 9. pp. 1658-1664, 2012.

[26] B. Mondal, K. Dasgupta and P. Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach," *Proc. Technol.*, vol. 4, pp. 783-789, 2012, doi: 10.1016/j.protcy.2012.05.128.

[27] N. Sran and N. Kaur, "Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing," *International Journal of Engineering Science Invention*, vol. 2, no. 1, pp. 60-63, January 2013.

[28] D. Babu L.D. and P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing*, vol. 13, no.°5, pp.2292–2303, May 2013, doi: 10.1016/j.asoc.2013.01.025.

[29] I. Fatima, N. Javaid, M. Nadeem Iqbal, I. Shafi, A. Anjum and U. Ullah Memon, "Integration of Cloud and Fog based Environment for Effective Resource Distribution in Smart Buildings," *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, pp. 60-64, doi: 10.1109/IWCMC.2018.8450422.

[30] S. Javaid, N. Javaid, S. K. Tayyaba, N. A. Sattar, B. Ruqia and M. Zahid, "Resource Allocation using Fog-2-Cloud based Environment for Smart Buildings," *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, pp. 1173-1177, doi: 10.1109/IWCMC.2018.8450331.

[31] A. Recioui, "Sidelobe Level Reduction in Linear Array Pattern Synthesis Using Particle Swarm Optimization," *Journal of Optimization Theory and Applications*, vol. 153, no. 2, pp 497-512, 2012, doi: 10.1007/s10957-011-9953-9.

[32] J. Wan, B. Chen, S. Wang, M. Xia, D. Li and C. Liu, "Fog Computing for Energy-Aware Load Balancing and Scheduling in Smart Factory," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4548-4556, Oct. 2018, doi: 10.1109/TII.2018.2818932.

[33] S. Zahoor, N. Javaid, A. Khan, B. Ruqia, F. Muhammad and M. Zahid, "A Cloud-Fog-Based Smart Grid Model for Efficient Resource Utilization," *Sustainability,* vol. 10, no. 6, 2079, 2018, doi: 10.3390/su10062079.

[34] Y. A. Ghani Alyouzbaki and M. F. Al-Rawi, "Novel load balancing approach based on ant colony optimization technique in cloud computing," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2320-2326, August 2021, doi: 10.11591/eei.v10i4.2947.

[35] D. Zhang, F. Haider, M. St-Hilaire and C. Makaya, "Model and Algorithms for the Planning of Fog Computing Networks," in *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3873-3884, April 2019, doi: 10.1109/JIOT.2019.2892940.

[36] K. Li, G. Xu, G. Zhao, Y. Dong and D. Wang, "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization," *2011 Sixth Annual Chinagrid Conference*, 2011, pp. 3-9, doi: 10.1109/ChinaGrid.2011.17.

[37] Q. Guo, "Task scheduling based on ant colony optimization in cloud environment," *AIP conference proceedings*, vol. 1834, no. 1, pp. 040039, Apr. 2017, doi: 10.1063/1.4981635.

[38] R. G. Babukarthik, R. Raju and P. Dhavachelvan, "Hybrid Algorithm for Job Scheduling: Combining the Benefits of ACO and Cuckoo Search," *Advances in Computing and Information Technology*, vol. 177, 2013, doi: 10.1007/978-3-642-31552-7_50.

[39] Z. Amini, M. Maeen, and M. R. Jahangir, "Providing a load balancing method based on dragonfly optimization algorithm for resource allocation in cloud computing," *International Journal of Networked and Distributed Computing*, vol. 6, no. 1, pp. 35-42, Jan. 2018, doi: 10.2991/ijndc.2018.6.1.4.

[40] S. Goyal *et al.*, "An Optimized Framework for Energy-Resource Allocation in a Cloud Environment based on the Whale Optimization Algorithm," *Sensors*, vol. 21, no. 5, 1583, 2021, doi: 10.3390/s21051583.

[41] T. S. Rani and S. Kannan, "Task Scheduling on Virtual Machines using BAT Strategy for Efficient Utilization of Resources in Cloud Environment," *International Journal of Applied Engineering Research*, vol. 12, no. 17, pp. 6663-6669. 2017.

[42] H. M. Mohammed, S. U. Umar and T. A. Rashid, "A Systematic and Meta-analysis Survey of Whale Optimization Algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, art. no. 8718571, 2019, doi: 10.1155/2019/8718571.

[43] M. N. Morshidi, I. Musirin, S. R. Abdul Rahim, M. R. Adzman, and M. H. Hussain, "Whale Optimization Algorithm Based Technique for Distributed Generation Installation in Distribution System," *Bulletin of Electrical Engineering and Informatics*, vol. 7, no. 3, pp. 442-449, 2018, doi: 10.11591/eei.v7i3.1276.

[44] I. S. Shahbudin *et al.*, "FACTS device installation in transmission system using whale optimization algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, March 2019, pp. 30-38, doi: 10.11591/eei.v8i1.1442

[45] M. W. Hassan and N. H. Abbas, "Controller design for underwater robotic vehicle based on improved whale optimization algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 609-618, 2021, doi: 10.11591/eei.v10i2.2288.

[46] D. C. Devi and V. R. Uthariaraj, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks," *The Scientific World Journal*, vol. 2016, art. no. 3896065, 2016, doi: 10.1155/2016/3896065.

[47] T. Kumrai, K. Ota, M. Dong, J. Kishigami and D. K. Sung, "Multiobjective Optimization in Cloud Brokering Systems for Connected Internet of Things," in *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 404-413, April 2017, doi: 10.1109/JIOT.2016.2565562.

[48] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software,* vol. 95, pp 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[49] S. R. Thennarasu, M. Selvam and K. Srihari, "A new whale optimizer for workflow scheduling in cloud computing environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3807-3814, March 2021, doi: 10.1007/s12652-020-01678-9.

[50] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, et N. Krasnogor, vol. 284, 2010, doi: 10.1007/978-3-642-12538-6_6.

[51] X. Cai, W. Li, L. Wang, Q. Kang, Q. Wu and X. Huang, "Bat algorithm with Gaussian walk for directing orbits of chaotic systems," *International Journal of Computing Science and Mathematics,* vol 5, no. 2, pp. 198-208, 2014, doi: 10.1504/IJCSM.2014.064070.

[52] R. F. Lopes, F. F. Costa, A. Oliveira and A. C. De C. Lima, "Algorithm based on particle swarm applied to electrical load scheduling in an industrial setting," *Energy*, vol. 147, pp. 1007-1015, 2018, doi: 10.1016/j.energy.2018.01.090.

BIOGRAPHIES OF AUTHORS

Afaf Saoud received her B.S and M.S degrees at the Institute of Electrical Engineering and Electronics (IGEE) University of Boumerdes, Algeria in 2013, 2015, respectively. She is currently a PhD student at IGEE and research member at the laboratory signals and systems. Her research interests are: data communication, smart grid, machine and deep learning. She can be contacted at: a.saoud@univ-boumerdes.dz.

Abdelmadjid Recioui is a professor at the Institute of Electrical Engineering and Electronics University of Boumerdes, Algeria. He obtained a PhD degree in electrical and electronic engineering option telecommunications from the Institute of Electrical Engineering and Electronics, University of Boumerdes in 2011. He holds also Master (Magister) degree in Electronic System Engineering which has been achieved at the Institute of Electrical Engineering and Electronics, University of Boumerdes in 2006. In June 2002, he finished his engineering studies at the institute of Electrical Engineering and Electronics, University of Boumerdes. He is a research Assistant at the laboratory signals and systems from January 2008 to present in Laboratory: signals and systems, Inst. of Electrical Engineering and electronics, University of Boumerdes. His research interests include: antennas, wireless communication systems, antenna array synthesis and design, capacity enhancement, system optimization, smart antennas, power system protection, power system optimization, power system communications. He can be contacted at email: a_recioui@univ-boumerdes.dz