

# MOCAB/HEFT algorithm of multi radio wireless communication improved achievement assessment

**Thaar Habeb Nasser, Ekhlas Kadhun Hamza, Ahmed Mudheher Hasan**

Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq

## Article Info

### Article history:

Received May 14, 2022

Revised Sep 21, 2022

Accepted Oct 10, 2022

### Keywords:

Broadcasting

Communication

HEFT

MOCAB

Multi-channel multi-radio

Simplicial complex

## ABSTRACT

Network-wide conveying is vital in remote associations, and the great part of these broadcasts are built on single-channel single-radio (SC-SR) network frameworks. The problem of the current work is divided into two parts. The first part shows that increasing broadcast and redundancy lead to an increase in time consumption. The second problem is solving complexity problems when tasks are scheduled in a heterogeneous manner in a computing system, where the processors in the network may not be identical and take different time periods to carry out the same task. The goals of this work are to reduce the total cost of network-wide broadcasting to minimize the search space and to solve the complexity problem when tasks are scheduled in a heterogeneous way in the computing system. The MOCAB algorithm is used to select the best transmission path over the network in the first stage. Then, the tasks will be scheduled using the heterogeneous earliest finish time (HEFT) algorithm to extract the values of actual finish time (AFT), earliest start time (EST), and earliest finish time (EFT). The performance of the MOCAB algorithm was evaluated with that of the HEFT algorithm in terms of the delivery ratio of packets delivered. The results showed that the MOCAB algorithm outperformed the HEFT.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Thaar Habeb Nasser

Department of Control and Systems Engineering, University of Technology

Baghdad, Iraq

Email: cse.20.24@grad.uotechnology.edu.iq

## 1. INTRODUCTION

Cloud computing is one of the most exciting professional disciplines in the current era. It has had an effect on businesses, information technology, program design, and data storage. According to the National Institute of Standards and Technology (NIST), cloud computing is a model for promoting access to pooled, accessible, on-demand, pervasive services that may be conveniently dispersed through various modes of service provider contact. [1]. A significant advantage of cloud computing is the avoidance of costly IT network maintenance costs. "A cloud is a type of parallel and expressed network made up of a collection of networked and virtualized PCs that are gradually supplied and implemented as at least one consolidated computing asset based on the level of administration determined by the administrator and clients," say [2], [3]. The cost of sending someone to settle or introduce an application is reduced to a minimum, and the organization saves money. As a result, employing cloud-based software is less expensive than purchasing a variety of them. Therefore, using a single multi-application cloud benefits everyone within the organization.

The cloud-based systems should be properly integrated with the business using the application programming interface that judges the application to be suitable with the organizations' goals. The firm does not have to spend money on this because cloud computing is constantly updated. Cloud computing can help businesses save money [4]–[6]. The cloud model is ideal for this purpose. Globally empty services must be

utilized to improve consumption rates and resource profits by increasing the economic productivity of these services. The primary goal of cloud computing is to exchange infrastructure and data with customers. It acts as a platform for its users to access apps and services. Cloud computing provides three types of applications: software as a service (SaaS), data network as a service (PaaS), and cloud infrastructure as a service (IaaS) [7]. The distributed system is cloud computing technology. Cloud computing requires the use of scattered resources that may be easily allocated or de-allocated [8]. The introduction of cloud computing as a new technology. It is rapidly expanding as a result of the massive expansion of internet services [9]. This technology distinguishes itself by providing services to consumers and allocating cloud resources to users [10]. The resources are compensated based on how much each user utilizes them provided typical scheduling methods, including asynchronous work management with a variety of queues for aperiodic and periodic activities. When estimating the number of resources required to schedule a collection of aperiodic jobs, both execution and data transmission costs were considered [11]. The major limitation for the optimization metric was the deadline by [12]–[22]. First, this review article provides a synopsis of the primary applications for field programmable gate array (FPGA-based) platforms in 5G networks and systems, taking advantage of the higher performance these devices bring [23]–[26]. Network slicers, cognitive radio systems, multiple input multiple output (MIMO) channel characterizers, cloud radio access network (C-RAN) accelerators, and network function virtualization (NFV)-based network slicers are among the primary applications that can benefit from FPGAs' high processing speed, power efficiency, and flexibility [27]. This research provides an examination of cutting-edge signal-quality and high-bandwidth transmission systems. This paper gives a complete assessment of the relevant literature, with a focus on improved transmission techniques in high-availability seamless redundancy (HSR) communications to achieve increased bandwidth efficiency and signal quality when compared to existing algorithms, the experimental results revealed that the proposed method could more effectively manage resources [28]. Advanced transmission systems include orthogonal frequency division multiplexing (OFDM), MIMO, and radio over fiber (RoF) [29]. The mechanisms for ranking tasks and backfilling were improved by extending the algorithms. In comparison to other algorithms, the findings of the algorithms primary domain controller (PDC) and direct cable connection (DCC) show a greater success rate and cost effectiveness. A resource provisioning technique was presented to help hybrid cloud infrastructures meet the deadline demands of data-intensive applications [30]. The task scheduling literature review is presented in Table 1.

Table 1. A review of the literature on task scheduling

Title	Parameters	Simulation tool	Enhancement	Drawback
“Task scheduling algorithms with multiple factors in cloud computing environment”	Measures of load the balancin' and cost	Cloud sim 2016 [8]	Improve the efficiency of the clunky algorithms.	Only\schek with\stuff algorithms doesn't look any metaheuri stic algorithm.
Symbiotic organism search optimization based task scheduling in cloud computing environment.”	Imbalance level, makespa time, and overall execution time	Cloud sim 2016 [2]	Reduce the level of imbalance and makespace to increase performance.	Consider task scheduling, but don'tjust concentrate on load balancing.
“Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud”	Degree of unbalance, duration of the makespan, and overall cost	Physical cloud environ mentusi Ng worksta Tion 2016 [9]	Reduced makespanti led to better overall performance	only one algorithm was compared (bee colony)

## 2. METHOD

In order to clarify what the idea of research is, the method has been divided into two main stages. The basic steps for each stage. Where the first stage is to choose the best transmission path over the network by applying the MOCAB algorithm. Then, the tasks will be scheduled using heterogeneous earliest finish time (HEFT) algorithm to extract the values.

The following is the simulation scene: Each node is positioned at random within a 1,000 m×1,000 m region. Four wireless radios are included in each node. 50 meters is the maximum wireless signal transmission range. The simulation configurations are displayed in Table 2. It is believed that the system has 12 accessible orthogonal channels. In order to maintain network connectivity and consistent channel distribution, the distributed greedy algorithm (DGA) strategy is utilized to assign channels to each radio of the node at the start of each simulation.

Table 2. Configurations for simulation

Configurations	Number
Size of workspace	1,000*1,000
Communication rance	50 m
Channel number	12
Radio number	4
N.Node	50 node

Actually, the main reasoning of such evolution is to reduce the interconnection in the adjacent sub-complicated. The hub might be output along the sets of potential repeaters if all of their neighbors are summed together for multi-operator core networks (MOCNS). The estimate has been completely reformatted so that there is no pivot in the pool of potential retransmissions. In this period, each center in the network will send messages as long as the entire organization is connected.

To improve network efficiency, there is a need to create alternate paths in the transmission. For instance, it is required to determine the second shortest way when the shortest path connecting the source and destination nodes is congested. If the shortest route is not accessible, it could be necessary to take the third shortest route. The MOCAB algorithms solves this problem to find more than one path in the network.

Stage two, scheduling tasks elite activities in parallel and distributed systems demand efficient consumption task scheduling. Static a directed acyclic graph (DAG) scheduling techniques are suggested in this thesis for a variety of circumstances They are designed for a limited number of processors and employ rundown scheduling techniques. The HEFT algorithm selects the job with the noticeable upward position at each step, distributes it to the best processor, and sets the soonest finish time limit using an inclusion-based technique. The following is the simulation scene.

### 2.1. Heterogeneous earliest finish time

Using a heuristic, a network of diverse workers is given a set of similar tasks while taking communication time into consideration. The inputs for HEFT consist of a number of tasks, each of which is represented as a DAG, a set of workers, a schedule for when each task should be executed on each worker, as well as a schedule for when each worker pair should be informed of the results of each job's execution. It comes from list scheduling approaches. A potential flow diagram for the network-aware HEFT approach is shown in Figure 1.

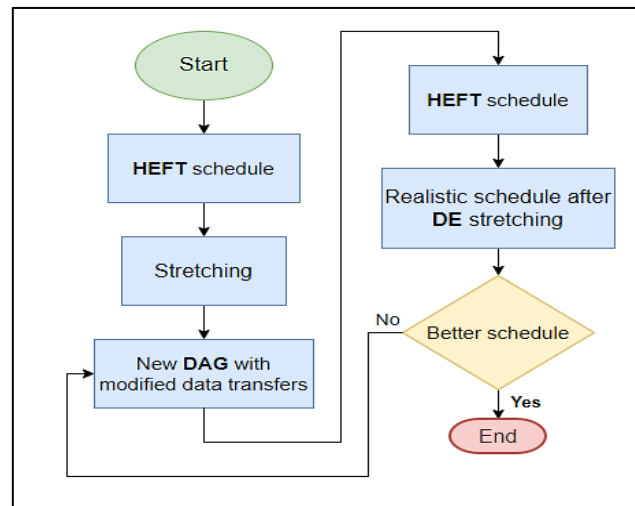


Figure 1. A flow chart of the network-aware HEFT method

Method (HEFT) a heuristic includes distributing a group of similar tasks among an array of workers while taking communication time into consideration. The strategy for developing a realistic makespan, network-aware timetable that accounts for all data transfers over shared bottlenecks is described in this section. The five steps in the flowchart for this methodology are shown in Figure 1. The following are these steps:

- In the first stage, a regular schedule is produced using HEFT on the DAG application.
- All data transfers that share a bottleneck are now picked out of the schedule and given the extensions to make them as long as they can reasonably.
- The final procedure includes changing the DAG by substituting the average communication costs anticipated in step 1 with all actual communication costs estimated in step 2.
- Standard HEFT is resumed after acquiring a DAG with specific adjusted communication costs. The main idea behind this step is that HEFT will improve on the plan created in step 1, with the expectation that the new schedule will be better than the previous schedule. Data transfer sharing bottlenecks will be implicitly considered throughout the HEFT ranking and mapping process due to altered communication costs. If, as in step 1, HEFT chooses to map tasks to computing resources in this phase, the realistic schedule generated at the end of step 2 will be used.
- In some circumstances, the data transfer times in the schedule created after step 4 may differ from those calculated in step 2 due to the changing way that data transfers interact with one another. For instance, it's possible that no bottleneck exists in the new schedule generated after step 4 for any of the flows that were stretched during step 2. As a result, more iterations have been carried out to try and determine the best time frame for the two-stage HEFT method:

a. The first phase: prioritizing tasks

Each task is given a priority during the early phase. Typically, the "upward rank" of each work determines its priority, which is defined recursively as follows. Ranking up (1) is [17] represented as:

$$R_{up}(N_i) = \overline{w}_i + \max(\overline{c}_{ij} + R_u(N_j)) \quad (1)$$

where  $n_i$  stands for the "displaystyle i"th task,  $(w_i)$  is the average cost of the job I calculation over all processors, and "displaystyle succ (n i)" is used. display style over line  $c_{i, j}$  ( $C(i, j)$ ) displaystyle  $n_i w_L$  is the average communication cost of the variables exchanged between tasks display style  $n_i$  and display style  $n_j$  between all pairs of workers. Succ (n i) is the set of all jobs that directly depend on task  $n_i$ . Keep in mind that the calculation of the "display style rank u (n i)" depends on how the ranks of all of its offspring are determined. The upward rank's goal is to depict the anticipated lag time between a task's completion and calculation. Different averages may yield different answers for averaged values like "displaystyle overline w i" ( $w_i$ ). It is possible to express rank (u(Nexit)) (2) as [17]:

$$\text{rank}_{u(\text{Nexit})} = \overline{w_{\text{exit}}} \quad (2)$$

$\overline{w_{\text{exit}}}$ =average Computation cost

rank down can be expressed (3) as [17]:

$$R_{down}(n_i) = \max \{R_d(N_i) + \overline{w}_j + \overline{C}_{ij}\} \quad (3)$$

b. The two phase: assigning tasks to workers

The second phase involves assigning tasks to employees. After assigning each task a priority, we now take into account all of the tasks and rank them according to importance. The assignment with the highest priority for which all dependent tasks have been finished is given to the worker who will finish it as soon as possible. The time it takes to make the processor available, compute the task on the worker, and send the necessary inputs to the worker all affect the completion time (it may be busy with another task). By using an insertion-based approach, HEFT fills in any gaps between already scheduled jobs that are the right size.

c. Earliest start time

The earliest start time (EST), it is referred to as the earliest possible time for an activity to start. It is calculated by going from the beginning to the last event in the network diagram. The processor  $p_j$ 's work  $v_i$ 's EFT, represented as  $EST_{vi, pj}$ , was calculated using (4). The formula for determining the EST is [19].

$$EST(n_i, p_i) = \text{MAX}\{\text{avail}[j], \text{AFT}(m) + \overline{C}(m, i)\} \quad (4)$$

where the time at which processor  $j$  is ready to start working on another job is called avail ( $j$ ), and the EST of the entrance task is always zero. Every central processing unit (CPU) has a separate availability period, with the exception of the beginning, when all processors' availability times are 0. The actual finish time (AFT) ( $m$ ) is the parent  $m$ 's AFT, and  $c_{m,i}$  is the parent  $m$ 's communication to task  $i$ . The maximum amount of time

between processor  $j$ 's availability time and the moment the required data came from all of the parents is the earliest task  $v_i$  start time on processor  $j$ .

d. Earliest finish time and actual finish time

The EFT it stands for the earliest possible moment that an activity can terminate. It is the EFT of job  $v_i$  on processor  $p_j$  and is represented as  $EFT(v_i, p_j)$ . It is acquired using (5). The EFT is shown with [18]:

$$EFT(n_i, p_i) = w_{i,j} + EST(n_i, p_i) \quad (5)$$

where the EFT of job  $v_i$  on processor  $p_j$  is formed by adding the ESTs and computational costs of task  $I$  on processor  $j$ . With Eq, the task's AFT may be determined (6). Actual completion time is given by [18]:

$$AFT(v_i, p_i) = \min(EFT(v_i, p_i)) \quad (6)$$

### 3. RESULTS AND DISCUSSION

MATLAB v2018b was used in the first stage to establish a network of 50 nodes with an area of  $1,000 \times 1,000$  m and random distances chosen by MATLAB. To improve network efficiency, we look for alternative paths if the main path becomes congested. Congestion, for example, can result in lower quality of service across the entire network, which can result in application crashes, dropped calls, data loss, reduced throughput, and dissatisfied customers. One of the most serious consequences of network congestion is data loss. Packet loss occurs when data packets are dropped during transmission, resulting in poor video quality, poor VoIP service, and online game stoppage. When the internet path is full, network congestion causes packet loss. After specifying the parameters, the workspace was chosen to be  $1,000 \times 1,000$  m in size, the node positions were randomly assigned (50 nodes were picked), and the euclidean distance between one node and the other was calculated. The original network graph objects were drawn. As shown in Figure 2.

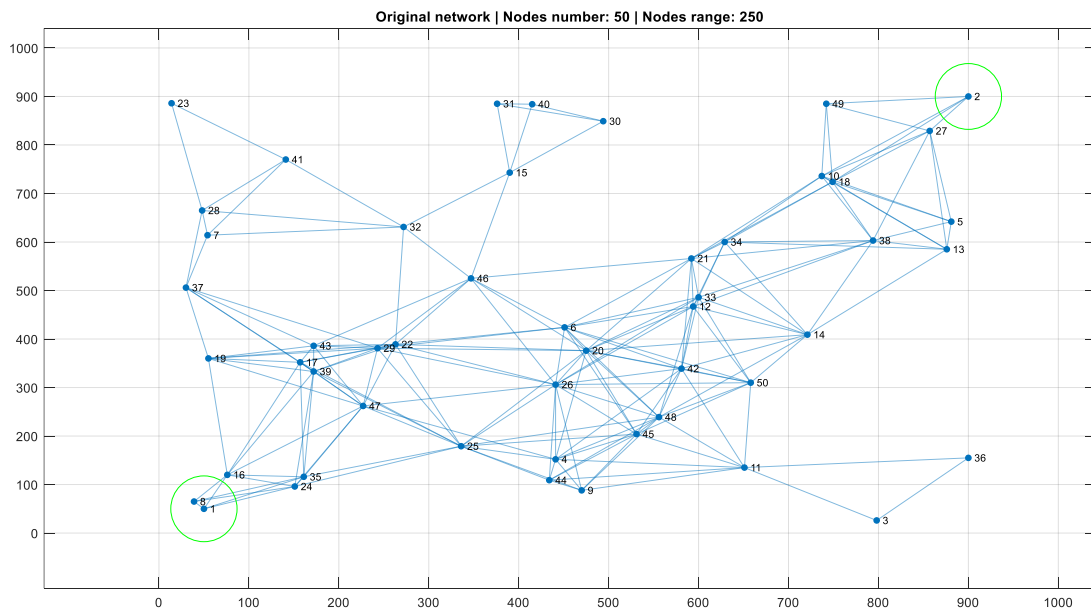


Figure 2. Original network

Then the packet is sent from the source node (N1) to the target node (N2) and the shortest main path in the network is calculated, which is colored green as shown in Figure 3 which passes through node (1,24,24,20,21,10,2). In this section, the proposed algorithm's performance evaluation and comparative analysis are covered for a variety of randomly created directed acyclic graphs (DAGs). The HEFT algorithm improves on the well-known scheduling algorithm when comparing the quantity of nodes to the closest time DAGs with various parameters. When compared to the suggested approach, the findings demonstrate that each node often achieves HEFT Reduced make-up for less nodes. However, as the number of nodes

increases, the results of the proposed algorithm are better. Display Figure 4 shows the (EST, EFT, and AFT) value, comparing the number of nodes with the time DAGs with different parameters in ten jobs. Figure 5 display packet delivery ratio analogy for N=25, among MOCAB/HEFT against MOCAB techniques.

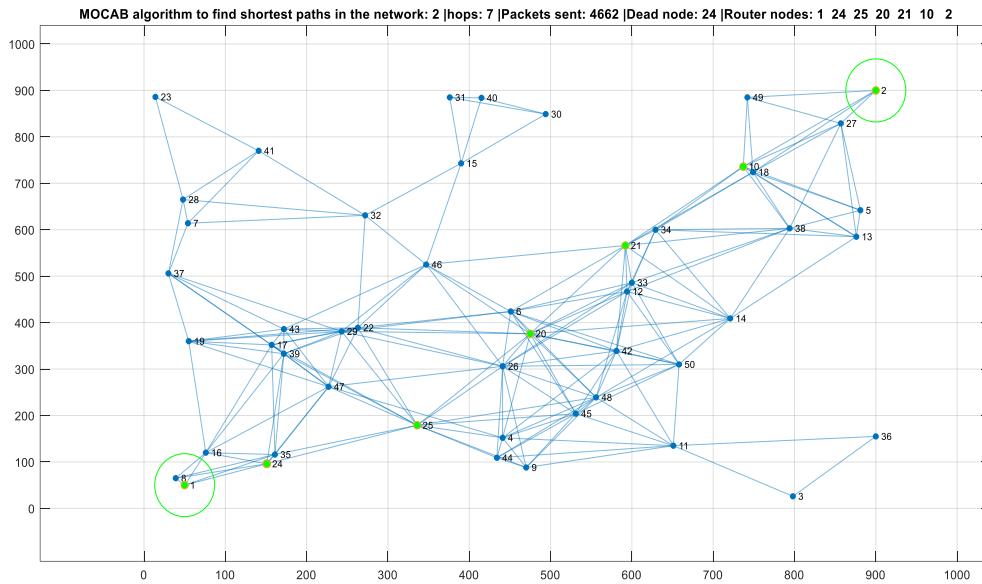


Figure 3. The shortest main path from the source node (N1) to the target node

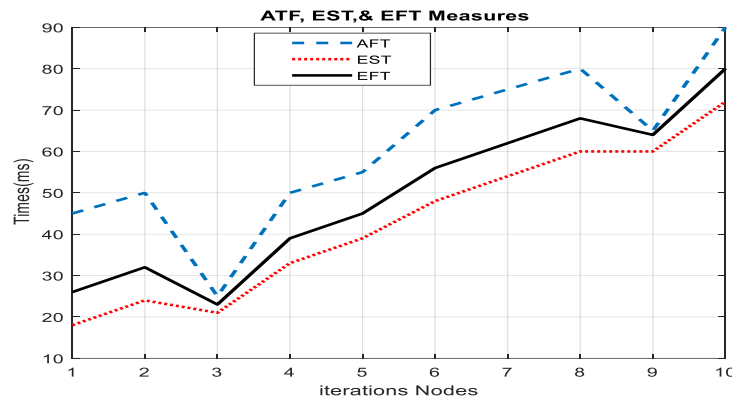


Figure 4. AFT, EST, and EFT for 10 Nodes examination using MOCAB/HEFT technique



Figure 5. Packet delivery ratio analogy for N=25, among MOCAB/HEFT against MOCAB techniques

#### 4. CONCLUSION

This work presented a methodology of the performance evaluation of multi radio wireless communication on MATLAB. This method is separated into two parts, with each stage containing specific steps to emulate. The first stage involved determining the shortest path through the network. The job scheduling for heterogeneous computing systems was demonstrated in the second step. The MOCAB algorithm was applied to find the shortest path between the nodes (the least expensive) and the alternative paths for the shortest path in the network were calculated. Isolated CAB, and MOCAB accomplishes an over 13% reducing in how much collecting per pack, that suggests less broadcasting accentuation. The HEFT algorithm was applied to calculate the AFT value, EST, EFT value and delivery ratio. In the future, this work could be expanded to include the following proposals. A node can be selected using the link standard, which is the fraction of retransmitted packets to total created packets. Consider task duplication at the processor selection step to reduce communications between processors and further reduce the makespan.

#### REFERENCES




- [1] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities," *Journal of Network and Computer Applications*, vol. 68, pp. 173–200, Jun. 2016, doi: 10.1016/j.jnca.2016.04.016.
- [2] M. Abdullahi, M. A. Ngadi, and S. M. Abdulhamid, "Symbiotic organism search optimization based task scheduling in cloud computing environment," *Future Generation Computer Systems*, vol. 56, pp. 640–650, Mar. 2016, doi: 10.1016/j.future.2015.08.006.
- [3] A. Thomas, G. Krishnalal, and V. P. J. Raj, "Credit based scheduling algorithm in cloud computing environment," *Procedia Computer Science*, vol. 46, pp. 913–920, 2015, doi: 10.1016/j.procs.2015.02.162.
- [4] E. H. Ali, A. H. Reja, and L. H. Abood, "Design hybrid filter technique for mixed noise reduction from synthetic aperture radar imagery," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1325–1331, Jun. 2022, doi: 10.11591/eei.v11i3.3708.
- [5] B. L. Pan, Y. P. Wang, H. X. Li, and J. Qian, "Task scheduling and resource allocation of cloud computing based on QoS," *Advanced Materials Research*, vol. 915, pp. 1382–1385, Apr. 2014, doi: 10.4028/www.scientific.net/AMR.915-916.1382.
- [6] S. W. Jirjees, A. M. Mahmood, and A. R. Nasser, "Passnumbers: An approach of graphical password authentication based on grid selection," *International Journal of Safety and Security Engineering*, vol. 12, no. 1, pp. 21–29, Feb. 2022, doi: 10.18280/ijssse.120103.
- [7] S. M. Abdulhamid, M. S. A. Latiff, and I. Idris, "Tasks scheduling technique using league championship algorithm for makespan minimization in IAAS cloud," Oct. 2015, [Online]. Available: <http://arxiv.org/abs/1510.03173>
- [8] N. Bansal, A. Awasthi, and S. Bansal, "Task scheduling algorithms with multiple factor in cloud computing environment," in *Information systems design and intelligent applications*, 2016, pp. 619–627, doi: 10.1007/978-81-322-2755-7\_64.
- [9] K. R. R. Babu and P. Samuel, "Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud," in *Innovations in bio-inspired computing and applications*, 2016, pp. 67–78, doi: 10.1007/978-3-319-28031-8\_6.
- [10] L. H. Abood, B. K. Oleiwi, and E. H. Ali, "Optimal backstepping controller for controlling DC motor speed," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 5, pp. 2564–2572, Oct. 2022, doi: 10.11591/eei.v11i5.3940.
- [11] K. D. Salman and E. K. Hamza, "Visible light fidelity technology: Survey," *Iraqi Journal of Computer, Communication, Control and System Engineering*, vol. 21, no. 2, pp. 1–15, Jun. 2021, doi: 10.33103/uot.ijccce.21.2.1.
- [12] D. Cui, Z. Peng, J. Xiong, B. Xu, and W. Lin, "A reinforcement learning-based mixed job scheduler scheme for grid or IaaS cloud," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1030–1039, Oct. 2020, doi: 10.1109/TCC.2017.2773078.
- [13] V. Arabnejad, K. Bubendorfer, and B. Ng, "Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources," *Future Generation Computer Systems*, vol. 75, pp. 348–364, Oct. 2017, doi: 10.1016/j.future.2017.01.002.
- [14] A. N. Toosi, R. O. Sinnott, and R. Buyya, "Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using Aneka," *Future Generation Computer Systems*, vol. 79, pp. 765–775, Feb. 2018, doi: 10.1016/j.future.2017.05.042.
- [15] C.-Y. Liu, C.-M. Zou, and P. Wu, "A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing," in *2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, Nov. 2014, pp. 68–72, doi: 10.1109/DCABES.2014.18.
- [16] H. Yang, "Improved ant colony algorithm based on PSO and its application on cloud computing resource scheduling," *Advanced Materials Research*, vol. 989–994, pp. 2192–2195, Jul. 2014, doi: 10.4028/www.scientific.net/AMR.989-994.2192.
- [17] M. Sulaiman, Z. Halim, M. Lebbah, M. Waqas, and S. Tu, "An evolutionary computing-based efficient hybrid task scheduling approach for heterogeneous computing environment," *Journal of Grid Computing*, vol. 19, no. 1, pp. 1–31, Mar. 2021, doi: 10.1007/s10723-021-09552-4.
- [18] M. H. Shirvani, "A new shuffled genetic-based task scheduling algorithm in heterogeneous distributed systems," *Journal of Advances in Computer Research*, vol. 9, no. 4, pp. 19–36, 2018, [Online]. Available: [http://jacr.iausari.ac.ir/article\\_660143.html](http://jacr.iausari.ac.ir/article_660143.html)
- [19] S. AlEbrahim and I. Ahmad, "Task scheduling for heterogeneous computing systems," *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2313–2338, Jun. 2017, doi: 10.1007/s11227-016-1917-2.
- [20] W. Sun, T. Peng, and Z. You, "Minimum redundancy broadcast in multi-channel multi-radio wireless networks," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, Dec. 2018, pp. 477–482, doi: 10.1109/CompComm.2018.8780917.
- [21] S. Al-Rubaye, E. Kadhum, Q. Ni, and A. Anpalagan, "Industrial internet of things driven by SDN platform for smart grid resiliency," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 267–277, Feb. 2019, doi: 10.1109/JIOT.2017.2734903.
- [22] B. K. Oleiwi, L. H. Abood, and A. K. Farhan, "Integrated different fingerprint identification and classification systems based deep learning," in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, Mar. 2022, pp. 188–193, doi: 10.1109/CSASE51777.2022.9759632.
- [23] A. Nasser and H. AL-Khazraji, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, pp. 721–730,






- Feb. 2022, doi: 10.11591/ijece.v12i1.pp721-730.
- [24] A. Mahmood and A. Kareem, "A new processing approach for scheduling time minimization in 5G-IoT networks," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 481–492, Jun. 2021, doi: 10.22266/ijies2021.0630.40.
- [25] A. G. Mahmoud, A. M. Hasan, and N. M. Hassan, "Convolutional neural networks framework for human hand gesture recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2223–2230, Aug. 2021, doi: 10.11591/eei.v10i4.2926.
- [26] H. J. Hassan and N. K. Hadi, "Implementation of wireless area network for patient monitoring system," *Iraqi Journal of Computers, Communication and Control & System Engineering (IJCCCE)*, vol. 17, no. 1, pp. 1–9, 2017.
- [27] A. H. Hadi and W. F. Shareef, "In-situ pipeline monitoring system design for identifying and locating damaging activities based on wireless sensor network," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 33–46, Apr. 2020, doi: 10.33103/uot.ijccce.20.2.5.
- [28] A. T. Namel, M. A. Sahib, and S. M. Hasan, "Bandwidth utilization prediction in LAN network using time series modeling," *Iraqi Journal of Computer, Communication, Control and System Engineering*, vol. 19, no. 2, pp. 78–89, Mar. 2019, doi: 10.33103/uot.ijccce.19.2.9.
- [29] P. Visconti, R. Velazquez, C. D.-V. Soto, and R. de Fazio, "FPGA based technical solutions for high throughput data processing and encryption for 5G communication: A review," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 4, pp. 1291–1306, Aug. 2021, doi: 10.12928/telkomnika.v19i4.18400.
- [30] A. Rafay, S. M. Idrus, K. M. Yusof, and S. H. Mohammad, "A survey on advanced transmission technologies for high bandwidth and good signal quality for high-speed railways," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 293–301, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp293-301.

## BIOGRAPHIES OF AUTHORS






**Thaar Habeb Nasser**    he received the Bachelor's degree in Computer Engineering from Al-Hussein University, Karbala, Iraq in 2014 and a master's student is currently in the Department of Control and Systems Engineering, Computer and Control Branch at the University of Technology, Baghdad, Iraq. He works a government employee to follow up on communications and electricity projects in the Department of Housing in the Holy Karbala Governorate. He can be contacted at email: cse.20.24@grad.uotechnology.edu.



**Ekhlas Kadhum Hamza**    she received her Ph.D in Electrical and Electronic Engineering from University Science Malaysia (USM), Malaysia in 2011. Currently, she is a assistant lecturer in Department of Control and Systems Engineering at University of Technology, Baghdad. Her main research interests lie in the area of communications and networking, including LTE system communications and networking, SDN platform, and wireless sensor. She is a member of IEEE. She can be contacted at email: 100374@uotechnology.edu.iq.



**Ahmed Mudheher Hasan**    he received the B.Sc. degree in Control and Systems Engineering (Computer Branch) from the Department of Control and Systems Engineering 2002, University of Technology, the M.Sc. degree in Computer Engineering from the same department 2006, the Ph.D. degree in Computer Engineering from the University Putra Malaysia, Malaysia. Currently, he working on hybrid intelligent systems with optimization techniques. He can be contacted at email: 60163@uotechnology.edu.iq.