

Research Article

Hybrid Genetic Algorithm for IOMT-Cloud Task Scheduling

Adedoyin A. Hussain ¹ and Fadi Al-Turjman ^{2,3}

¹Computer Engineering Dept and Research Centre for AI and IoT, Near East University, Nicosia, Mersin 10, Turkey

²Artificial Intelligence Engineering Dept and AI and Robotics Institutes, Near East University, Nicosia, Mersin 10, Turkey

³Research Centre for AI and IoT, Faculty of Engineering, University of Kyrenia, Kyrenia, Mersin 10, Turkey

Correspondence should be addressed to Adedoyin A. Hussain; hussaindoyin@gmail.com

Received 21 March 2022; Revised 6 June 2022; Accepted 16 August 2022; Published 5 September 2022

Academic Editor: Ashish Bagwari

Copyright © 2022 Adedoyin A. Hussain and Fadi Al-Turjman. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Task scheduling for the cloud is one of the main advances in IoMT stage, which impacts the whole execution of the cloud resource. Cloud is a proficient headway for computation, and it encompasses data storage, management, and manipulation in large volumes. Thus, a proposition is being made a better approach to proffer task scheduling in the cloud. In this case, a new hybrid genetic algorithm (HGA) is proposed. The proposed HGA method will be justified by contrasting it with the previous researches and approaches. The CloudSim is utilized to quantify their effect on various metrics like timing factors and resource utilization. The proposed HGA technique enhanced the viability of task scheduling with a better execution rate of 32.57 ms. Thus, the experimented outcomes show that the HGA also reduces cost profoundly.

1. Introduction

A reasonable scheduling technique is required to schedule these IoMT requests to cloud resources. Scheduling task is classed as one of the focal issues for computing in IoMT-cloud. The IoMT-cloud is progressed with the improvement of PC and association advancement. This prompts the execution of all tasks efficiently and also provides patients with formidable QoS [1, 2]. Task scheduling for the cloud is one of the main advances in IoMT stage, which impacts the whole execution of the cloud resource. Authors in [3] proposed a scheduling computation for tending to the cloud task to further develop scheduling estimation, which can get the more unobtrusive time and lower cost for doing for each process. Numerous investigations show that IoMT-cloud task scheduling problem is termed as a NP-hard problem, which has been concentrated by various analysts. The work in [4] proposed particle swarm optimization (PSO) to deal with the idea of the organization of users. It has achieved extraordinary results in the field of arranging resources to cloud tasks ensuing in finishing a huge number of coherent tasks. It incorporates a moderate speed of processing and is essentially caught in more waiting time.

Authors in [5] prepared a hereditary reenacted tempering estimation for task arrangement with twofold fitness, and this can effectively change the solicitations of the clients for the properties of tasks and work on the clients' satisfaction appropriately. Authors in [6] use the procedure for tending to the cloud task scheduling by exceptional self-changing underground ant colony optimization (ACO) in handling the scheduling of tasks.

To work on scheduling issues adequately in the IoMT-cloud stage, the environment has to be viewed and studied. Figure 1 provides a proper view of the IoMT-cloud. Cloud is a proficient headway for computation, and it encompasses data storage, management, and manipulation in large volumes and uses that data to understand a given outcome [7–11]. This reduces the outright period of manpower and lessens the cost, in the health system. This is a foremost advancement that makes use of the probability of business execution of computer programming with patients publicly [12]. IoMT-cloud is another progression gotten from grid computing, and it suggests involving enrolling assets in an association and accommodating recipients on demand through the Internet [13]. Scheduling in the cloud is one of the major factors in IoMT that it is considered to be the

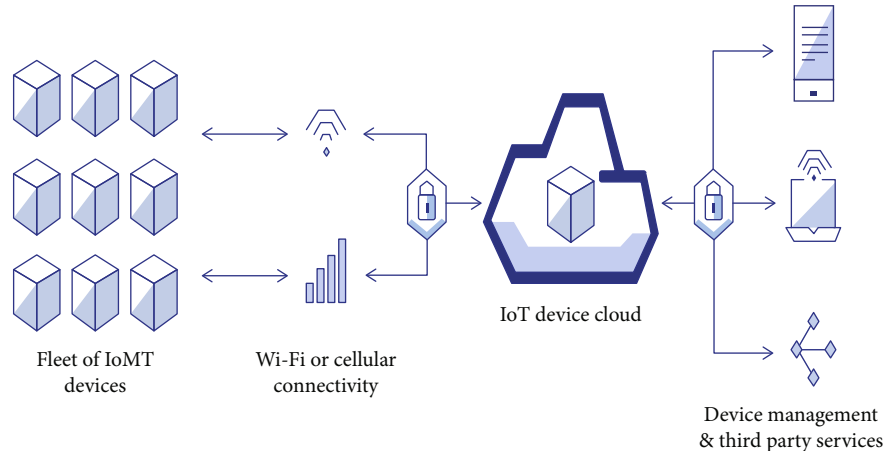


FIGURE 1: IoMT-cloud platform illustration.

essential factor that controls several operations like flexibility, patient resource sharing, and power use. Regardless, there are various troubles normal in IoMT scheduling. High execution rate can be caused by the scheduling technique, and task weights for each process will be scattered across all resources adequately and effectively to get less hold-up time, execution time, and most outrageous throughput. This process can solve a segment of the troubles faced in IoMT computing.

The critical ideal process of IoMT experimentation is that it propels authentic use of resources [14]. Each impacts the other. Fitting these IoMT tasks adequately might achieve efficient utilization of resources. With this, patients can get content wherever and without hoping to contemplate the working of the establishment. IoMT works on no limit provided that there is an internet connection. Therefore, task sharing and resource utilization in the IoMT stage are two sides of a lone coin. The cloud propels organizations to breach the gap between users or patients [15]. Cloud organization can scale up or down resources in the IoMT stage, per the solicitations of the applications. The cloud organization client can rent the resources at whatever point and release them with no difficulty. The cloud organization provides remote assistance regarding any application or resource to the users. This is a central purpose of the IoMT-cloud computation; nonetheless, the organization may be responsible for paying additional costs for this proposition. The example of the IoMT-cloud trends is depicted in Figure 2. Consequently, resource management and task scheduling are required bits of IoMT-cloud research [16]. In handling complex task scheduling-related issues, the usage of scheduling computation is recommended. The adequacy of resource uses depends upon the scheduling and resource weight, rather than the unpredictable designation of resources. Scheduling in IoMT-cloud is for the most part used for handling complex endeavors (client requests). Such arranging computations impact the resources.

In this work, the commitment provided has described major factors that are required for task scheduling, which are as follows: a survey on task scheduling optimization; a portrayal and analysis of the result gotten from the examina-

tion; proposing an HGA for IoMT-cloud compared to previous studies; and summing major points and issues in this paper.

This work is segmented as follows. Section 2 gives the background about scheduling procedures and techniques in the IoMT-cloud stage. An introduction to various literature reviews which add to the idea of the method and experimentation utilized is introduced in Section 3. Section 4 provides the problem statement proposed in the research, while in Section 5, the used technique, materials, and the proposed method utilized in experimenting are discussed. Section 6 discusses the outcomes of the experiment. Section 7 presents the conclusion and the closing remarks. Table 1 shows the list of abbreviations used.

2. Background

Here, it discusses the notable optimization scheduling techniques. More light will be given in the literature concerning related works in the cloud environment.

2.1. Shortest Job First (SJF) in IoMT-Cloud. In this conventional methodology, a need is given the length of the task process. It begins from the least to the task with the highest process. In this model, the task is organized on their necessities. The mentioned resource is then allocated to the task process that has the littlest time [18]. It is a rule of a medium waiting time among all other computations. The model is known as a precautionary methodology that picks on cycles that have the least execution time. It does not guarantee task fairness when tasks are distributed to VM [19]. Be that as it may, it has a more drawn out finish time. With this, this procedure is said to be a static scheduling procedure. This is a direct result of tasks with high processes being left unattended to, while little processes are taken care of. It has these processive traits:

- (i) It will always be aware of the next task process
- (ii) It lessens the waiting time for the task process as it processes little task before huge ones

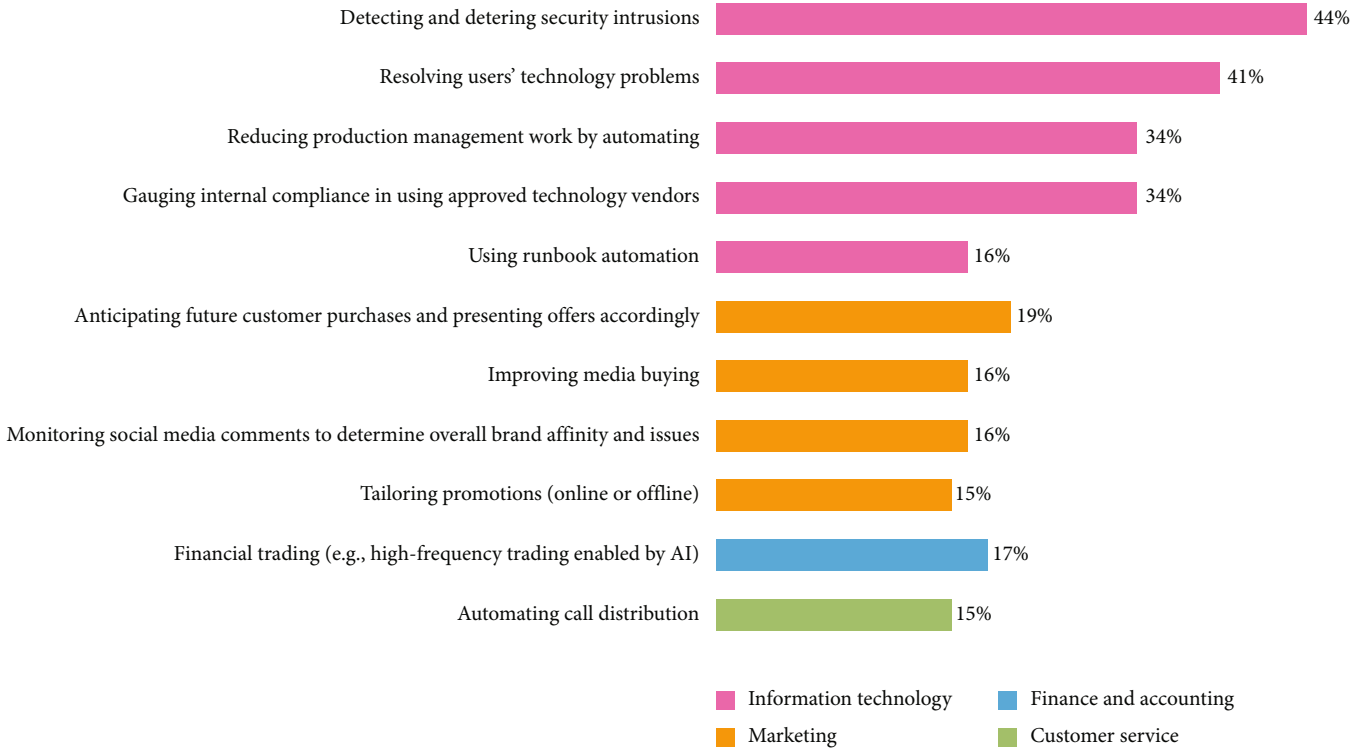


FIGURE 2: Trends of IoMT-cloud [17].

TABLE 1: Abbreviations used in the work.

Terms	Meaning
SJF	Shortest job first
PSO	Particle swarm optimization
ICT	Information and communication technologies
QoS	Quality of service
RR	Round robin
IaaS	Infrastructure as a service
TET	Total execution time
AI	Artificial intelligence
ML	Machine learning
NP	Nondeterministic polynomial
GA	Genetic algorithm
ACO	Ant colony optimization
TFT	Total finish time
TWT	Total waiting time
PaaS	Platform as a service
FCFS	First come first serve
PC	Personal computer

2.2. *First Come First Serve (FCFS) in IoMT-Cloud.* FCFS is a customary methodology, a task that shows up first is served first. The latest request from the patients is installed into the tail of the line. The solicitation of assets relies upon the time of task arrival. This is one of the standard methodologies, and it is more alluring than different methodologies [20]. It depends upon the standard of FIFO with lesser complexity

than other computations techniques [21]. This process is immediate and expedient. Whenever we have immense requests, all requests delay until the primary occupation is done. To evaluate the achievement by this technique, we will test them and subsequently gauge their impact on a few legitimate rules in the methodology. With this, this booking strategy is the static methodology. The FCFS has these qualities:

- (i) Prioritization depends on the main request and each cycle towards the end finish before new cycle
- (ii) This kind of computation does not work honorably with postponing traffic as holding time and mapping are for the most part on the higher side

2.3. *Round Robin (RR) in IoMT-Cloud.* In this conventional methodology, all task process is executed with fairness. In a general sense, this approach is differentiated to the static type as a result of its dynamicity [22]. Right away, all task processes are given equivalent time for execution which is once in a while called the quantum time. All processes are kept in the solicitation as they show up [23]. In light of the model utilized in this work, the quantum is picked given the mean of the cumulative process time. Right after deciding the mean, it will portray the finish time at the same time. It usually has these properties:

- (i) Assuming we apply a more restricted quantum, by then, productivity might become low
- (ii) Juggling the quantum to get a decent time will increase time process efficiency

2.4. Genetic Algorithm (GA) in IoMT-Cloud. The GA is an AI strategy that has gained ascend in execution lately. The GA is a metaheuristic approach that deals with the foundations of hereditary qualities and regular determination. The GA approach begins in light of its underlying population [24]. The general population is taken self-assertively to fill in as the early phase for this procedure. A fitness calculation is always used to get the fittest of the chromosome for a general population. Given these factors, chromosomes are picked, and mating operations are carried out on them for the new generational population. The fitness variable surveys the idea of each successor [25]. This paper will utilize an HGA approach which is an adjusted GA approach for greater legitimacy. It will be examined further in the next segment. The fundamental GA approach is exhibited below:

- (i) Initialization: Generates an initial populace
- (ii) Fitness: Based on the fitness value, calculate for each chromosome
- (iii) Mating pool: Select the 2 best chromosomes after wellness handling, and this is otherwise called the guardians. Hybrid produces results by choosing chromosomes to play out this activity to deliver new chromosomes known as posterity. At long last, mutation happens by playing out the change strategy on the chromosomes for a superior chromosome
- (iv) Fitness: Based on the fitness value, calculate for each chromosome
- (v) Repeat 2 to 5 until meeting the end condition. A stopping condition may be the number of cycles
- (vi) End procedure by giving the result of the best chromosome as the last outcome

3. Literature Review

This section gives an overview of several studies on scheduling arrangement and resource distribution. Various experts put forth replies to solve the problem of scheduling. Authors in [26] put forth a multiobject technique that applies better differential progression computation. However, task types are not emphasized in this philosophy. Thus, further improvement can even presently be made. However, this current technique provides a cost and time model for conveyed scheduling. This process does not depict the genuine utilization of resources. Authors in [27] put forth a queue arranging and changing estimation that does not emphasis on work sizes. A programming nonlinear model was used to disperse assets for tasks. Likewise, in [28], they introduced the preparation of tasks reliant upon an excursion lining model. Nonetheless, writers in [29] proposed the scheduling of users request while pondering transmission of information as a resource. In [30], they proposed analytic hierarchy process (AHP) situating based endeavor arrangement. The proposed system does not focus on rashly finishing the processes and starvation. Authors in [31] proposed an

acquainted moving skyline approach with planned tasks. They considered the FCFS process for managing demands when assets are free. Subsequently, in [32], they anticipated equivalent extraordinary weights based on incoming demands. Writers in [33] put forth a need-based business scheduling estimation for use in disseminated registering. Authors in [34] put forth the use of a metaheuristic upgrade to diminish costs through task arranging. In [35], they introduced the high-level cost of energy and coating delay goals. This system does not ponder the availability of resources or the weight of tasks. Be that as it may, in [36], they proposed the usage of modified bug area upgrade in load changing. Authors in [37] proposed a system subject to a multiguidelines computation for arranging specialist load. This strategy works on the make-span of a work. Thus, in [38], they put forth a resource assignment problem that means to restrict the full-scale energy cost of appropriated scheduling structures while meeting the foreordained client level SLAs according to a probabilistic point of view [39–41]. Here, they have applied a contrary philosophy that applies a disciplined approach on the off chance that the client does not meet the SLA plans. Consequently, in [42], they proposed a structure subject to the requirement for performing a distinguishable weight schedule that uses coherent movement measures. The technique robotizes the cycle and diminishes the piece of human management, while in [43], they introduced a central weight changing the decision model in the cloud. A couple of makers have proposed a heuristic estimation to handle task arranging and resource task issues portrayed already. Regardless, the technique is lacking in choosing the weight of center points and, arrangement nuances, and the complete phase has no support, as needs are achieving a lone reason for dissatisfaction. Moreover, in [44, 45], they focused on arranging endeavors while contemplating various goals. This technique coordinates additional examination on task planning and resource distribution.

Another approach scheduling approach is using the ant colony optimization (ACO) planning computation. Authors in [46] proposed using hybridization of bug region movement strategy for reasonable weight-changing process, using bug settlement min-max methodology, and inherited estimation. This, finally, processes the amount of pattern of virtual machines from the cloud applications. They proposed a solid method to restrict movement cost of VM and also hold tight to the SLA (service level agreement) which guarantees a QoS. Through this, the need is apportioned to VM to extend the response period of the system and to achieve better weight changing. Authors in [47] proposed a procedure that assists the starvation in work change. To vanquish this trouble, they used innate computation with the logarithmic least square strategy. With this, [48] put forth an improved GA by using the fragmented people decline procedure planned parenthood of the rocky mountains (PPRM). Authors in [49] have audited keen cloud scheduling for load changing and proposed antlion optimizer (ALO) to provide better outcome in changing the cloud storage. After this cycle, GA is implemented to the new populace and observes fitness

regard. This gives more huge courses of action, while ALO handles the gigantic issue in space.

The following are the three principle exercises. Fundamental GA has terms called mutation, crossover, and fitness work. Authors in [50] have looked into extraordinary GA for making a response. They have considered a need-based basic evaluation. By this idea, they achieved better ordinary response time and augmentations cloudlets with change encoding. It helps with decreasing time in waiting. In [51], they have proposed a cloud-based approach for the most part of the storage and dynamic multimedia load balancing (CSdynMLB) technique to change the stack for specialists. They have introduced job unit vector (JUV) and processing unit vector (PUV) terms to get the fitness of individuals. A similar need is applied to every one of the requesting and ensures better QoS, high interoperability, and flexibility. Authors in [52] have proposed cross variety genetic computation like genetic ant colony algorithm-virtual machine placement (GA-GEL) estimation for VM load balance process in the cloud. In [53], they put forth the genetic ant colony algorithm-virtual machine placement (GACA-VMP) method for managing settling VMP issues using further developed ACA. The outcome showed up with the Cloud Analyst proliferation gadget that fluctuates with a different number of server ranches. Through this procedure, they have picked a feasible way in two phases. This is gained to successfully pick the genuine specialist and assemble the resources [54–59]. Yet it has been discussed in the literature that there are still various areas that need tending to, and this work proposed here aimed to settle these issues.

4. Problem Formulation

This research addresses the issue of task scheduling in the IoMT-cloud which is a widely distributed and heterogeneous environment. Here, the sets of processors and tasks are considered as P_m and T_n , respectively. Let us say the available P_m processors for some set of tasks T_n , with no sharing during execution. Let ECT be the expected completion time, which contains estimated time for execution of a particular task on each resource, and the estimated completion time of a resource. The goal here is to reduce the total completion time of task execution. To increase resources utilization and minimize the time, the tasks have to be efficiently scheduled or mapped appropriately on the resources available. The depiction is shown in Figure 3.

$$\begin{aligned} T_n &= (T_1, T_2, \dots, T_n), \\ P_m &= (P_1, P_2, \dots, P_m), \end{aligned} \quad (1)$$

where T_n is given by set of tasks and P_m is set of resources.

The goal here is to map $T_n \longrightarrow P_m$.

5. Proposed Methodology

The IoMT-cloud has different characteristics which gives benefits to the end client. The major features of IoMT-cloud are self-redesigned, adaptability, and customization.

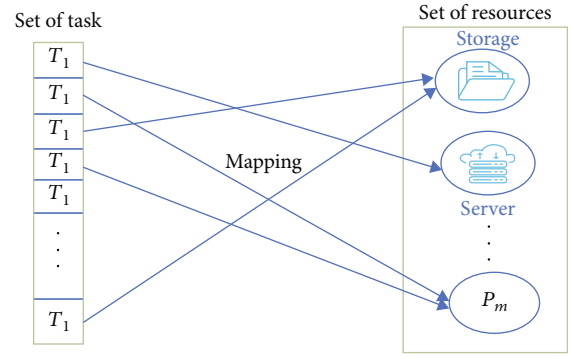


FIGURE 3: Scheduling task mapping.

The structure intends to work on the display of scheduling in IoMT, while simultaneously diminishing computational costs. The hopeful features of cloud resources are essential to permit organizations that absolutely layout clients' fulfillment. The key objective is to expect the best technique for the scheduling process when required. Certain bodies should be considered while satisfying these destinations like cloud providers and clients of the cloud. To achieve this, we play out a calculated assessment for scheduling in the IoMT-cloud environment and optimize it by utilizing the proposed AI approach which will be the HGA. Furthermore, we separate the essentials and consequences of utilizing quality of service (QoS) with the proposed outcome. The calculation ought to be sufficiently skilled to manage the issues related to scheduling like resource questions, lack of resources, and over-provisioning of resources.

The user demands the assets, and the cloud supplier is liable for the task of the expected asset, so the client evades the infringement of the service level agreement (SLA). For the strategies for arranging IoMT-cloud assets, the cloud information service (CIS) is responsible for the properties of each resources and its availability. The cloud scheduler must be efficient to designate different virtual machines (VMs) to various processes. Thus, the scheduling process in the IoMT-cloud is shown in Figure 4. The proposed AI technique will use a hybrid genetic algorithm (HGA) with the blend of a dynamic round robin and a local search, with a variation in step from the authors' previous research, and the depicted outcome will predict the result by recognizing the one with the best result. The outcomes are broken down in light of various related limits (the client and supplier desired) with the best outcome being discussed in the accompanying subsection.

5.1. Hybrid Genetic Algorithm (HGA). GA portrays a general population upgrade technique in light of a progression pattern of nature. In GA, each chromosome addresses a possible response for an issue and is made from a progression of characteristics. Given fitness factors, chromosomes are picked, and mating is performed for a new populace to emerge. The fitness evaluates the idea of each successor. Fitness is described to look at the worth of the chromosome for the general population. The cycle of fitness calculation is

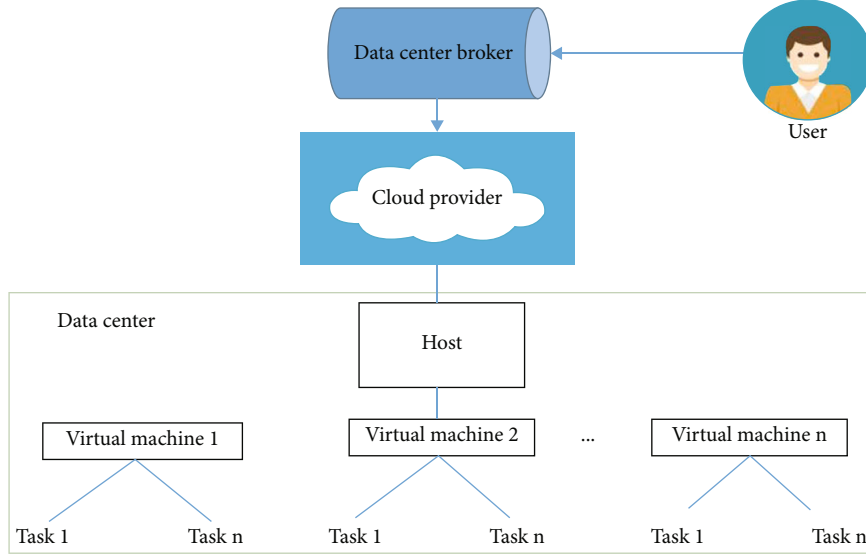


FIGURE 4: Task scheduling process structure.

repeated until satisfactory successors are made. Here, this approach will have a slight variation from the author's previous research. The proposed HGA will combine dynamic round robin and a local search algorithm known as hill climbing. The flowchart of the HGA in the IoMT-cloud is displayed in Figure 5. The HGA in the IoMT-cloud process is shown as follows:

(i) Initializing the Process

Introductory generation of populace P consisting of chromosomes. In this scheduling problem, we are using the datasets that have been taken from various IoMT devices from users' requests as input. The cumulative time to complete all the operations on all machines will be considered for the IoMT devices. The main objective of the problem is to find a valid schedule that yields the minimum completion time.

For initializing the initial population, the individuals in the population will consist of task and VM ID. This will be embedded together to form a chromosome, and each chromosome is a solution on its own. Each chromosome will have a representation like this: (e.g., VM2: - TS1-TS3-TS6).

(ii) Dynamic Round Robin and Fitness Calculation

In this mode, the round robin will work on a dynamic quantum time. The quantum time will be the median of all the processes. Let us consider the processes (T4, T5, T6, T7, and T8) with their respective completion times of (10, 5, 5, 5, and 10); in this case, the quantum time will be given has the median of these processes. Implementing this will grant task fairness for the task with longer and minima time process. This procedure will continue until all the processes are executed.

Thus, after the dynamic round-robin process, the fitness can be calculated. The completion time for task T_n on R_m is

given using

$$TCT = \max (CT_{n,m}), \quad (2)$$

where $\max (CT_{n,m})$ is the maximum time to complete task T_n on R_m . T_n and R_m are set of tasks and resources, respectively. m and n are the numbers of virtual machines and tasks. TCT is the total completion time.

Then, to minimize the completion time TCT, the execution time of every task for every VM must be calculated. The processing time is to be calculated where P_{nm} is the processing time for task P_n on R_m and C_n computational complexity of task P_n and the processing speed of the virtual machine is PS_m .

$$P_{nm} = \frac{C_n}{PS_m}. \quad (3)$$

After getting the processing time, the processing time of every task in the VM has to be calculated using P_j

$$P_m = \sum_{i=1}^n P_{nm}. \quad (4)$$

(iii) Selection

Once the fitness is calculated for each individual or chromosome, tournament selection is utilized to select the better chromosome from the pool of chromosomes. These selected chromosomes are used to perform crossover and mutation operations. This selected chromosome will be the parents. Chromosomes are selected, and the fitness is compared and then whichever chromosome possesses a lesser completion time is the best chromosome.

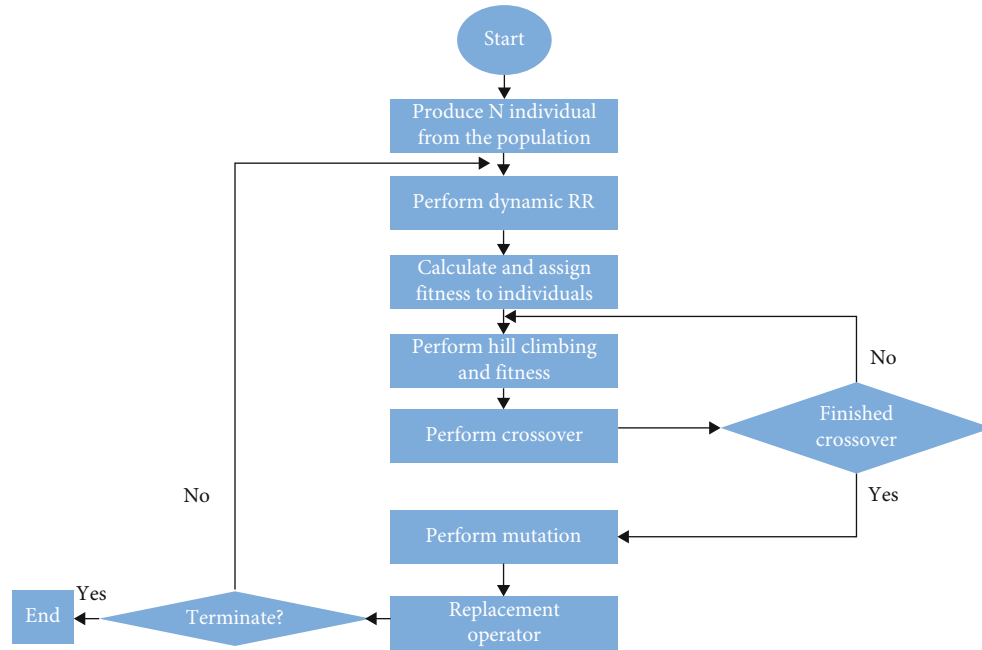


FIGURE 5: HGA flowchart.

(iv) Hill Climbing Operation

The newly generated parents will be used to perform the hill climbing operation. The hill climbing is going to be a stochastic approach where the initial hill point for the chromosome is chosen at random towards the uphill move. It is an increasing value mode. It generates new solutions on the hill based on its search space. The probability of new solutions might vary due to the steepness of the hill. The hill climbing will consist of two main approaches. A candidate generator is one that maps a solution to a set of possible successors and the evaluation criteria to rank every valid solution. The process will assist to generate a more fit parent that can produce a better offspring.

(v) Crossover and Mutation

This operation is also referred to as the mating pool. The parents get from the selection operation will be used to perform the crossover. Here, uniform crossover is applied. After the crossover, two new chromosomes will be produced. These two new chromosomes will make it four chromosomes in total. From the four new chromosomes, the best of these will be selected as the new offspring, and the latter will be added back into the population for possible selection later on. After this process, the mutation operation will be applied for a fitter value.

(vi) Replacement

Update the populace P. This will replace the populace with better chromosomes from the new generation of offspring. Repeat stages 2 to 6 until stopping criteria are met.

- (i) The resulting output will be the best chromosome

(ii) End process

5.2. *Experimental Process.* Assumptions to be viewed while planning the process in the IoMT-cloud are as follows:

- (i) Each task is dispensed to only a solitary VM resource
- (ii) The task will be greater than the amount of VMs. This infers that every VM ought to process more than one task
- (iii) The task is not obstructed once their executions start
- (iv) The lengths of the task will be of various sizes
- (v) The available VMs are of prohibitive use and cannot be split among different tasks. It suggests that the available VMs cannot consider various tasks not until the realization of the present task is in progression
- (vi) VMs are independent concerning resources and control

5.3. *Visualization.* The huge motivation driving depiction is portraying the information and graphically speaking with it. This is with the creative aspect that the experimental outcomes are portrayed graphically. The case of information understanding is portrayed as processing and manipulating data, information depiction, and construction attestation, outcome depiction, and finally looking at the information. The yield will be depicted visually in this work for more understanding.

5.4. *Computational Environment.* Eclipse is an environment for data evaluation and authentic approaches. The assessments were implemented using this IDE. It is an open-source software which implements the use of AI methodologies. CloudSim is used for simulating in the IoMT-cloud stage. Java programming language is likewise the most outstanding language, and it offers various library packages that can handle information science attempts, for example, information assessment, information predealing, and explicitly, working of different techniques. The research is implemented using a PC with Intel i7 processors: 2.3Ghz, GPU: GEFORCE, Disk: 1 TB, RAM: 12GB.

6. Results

Each experimented model and the proposed model will be tested to anticipate which model gets the higher assessment result. To assess the plausibility of our technique, the proposed technique has been contrasted on various optimization and hybrid approaches. The models have been endeavored with various settings to accomplish the most fundamental TWT, TET, TFT, cost, energy efficiency, and resource utilization. This work has done a lot of different assessments with the most reassuring scheduling computations. This work has used traditional optimization and other hybrid algorithms for contrasting with our model to outperform the communicated scheduling issue in IoMT-cloud and accordingly improve it with the proposed model. Likewise, various VMs were used, and various IoMT tasks are used in this evaluation. Each model shows its capability while scheduling. Each model used a relative region of educational collections. Right after the best model is displayed, we see its usefulness with the recently referenced qualities to best predict these outcomes. As follows the eventual outcome of the models is clarified in this part. Eclipse and CloudSim were used which include different libraries for this task.

6.1. *Metrics and Parameters.* For validating the results of our proposed techniques with other models, the computational metrics below are used for this work. The authors add one more metrics which is energy efficiency in contrast to their previous work, though Table 2 depicts the parameters utilized.

TWT (total waiting time): This is a user-desired criterion. It is the wait time for task execution when a couple of resources compete for a particular resource. This time is the time spent waiting by the cycle or errand on the queue waiting for execution.

TET (total execution time): This is a user-desired criterion. The proportion of time to execute a cycle is a basic part. This time intimates the time between the appearance of a process and the finish time. This is likewise the aggregate sum of time spent by the cycle from coming in the queue until it finishes.

TFT (total finish time): This is a user-desired criterion. It is the distance on schedule from the beginning of an assignment until it wraps up. This is the all-out time at which an undertaking finishes its execution.

TABLE 2: Used parameter.

Parameters	Value
Task	10-40
Data center	0-3
Population size	120
Iteration	100
Mutation rate	0.05
Crossover rate	0.6
Data center	0-3
Bandwidth (mbps)	500-1000
Ram (Mb)	512-1024
Machine	0-14
Processing elements per Vm (Mips)	500 - 1000

Resource utilization: This is a service provider desired criterion. The utilization of resources is one more parameter that depicts the amplification of assets used. The usage of resources ought to be high in the scheduling framework, though providers need to attain maxima profit by rendering a set number of resources. This parameter is one of the primary meanings in task scheduling. The resource will be kept occupied. Also, reaction time and throughput are huge; however, one more significant boundary for task execution is the utilization of resources.

Average resource utilization

$$= \frac{\text{Time is taken by resource } i \text{ to finish all the task}}{\text{Makespan}} \times n. \quad (5)$$

Status/availability: This defines the resources that are available at a given time. This is a huge element in closing how to scatter and apportion the right assets for a given VM. Resource availability is one of the principal parts of scheduling. The accessibility status is a triumph when the right resource is being consigned to the VM.

Throughput: This is a service provider desired criterion. Throughput can be portrayed as the extent of a process being completed per time unit. Thus, throughput is the cycles executed over jobs completed in a unit of time. The schedule should want to extend the quantity of tasks executed per time unit.

Energy efficiency: Energy consumption is the power consumed during the processing of each client's request. To attain energy efficiency, the consumption of power must be reduced drastically. This is one of the major factors to be considered to arrive at a greener environment.

Cost: This is the monetary cost that depicts the total aggregate that ought to be paid by the client for the asset being utilized. This monetary cost will be established on how much time is spent by the client on a particular asset. The equation below portrays how it is calculated where T alludes to the time the resource being utilized and C infers the money-related cost of the resource per time unit. The

price of each resource is depicted in Table 3.

$$\text{Cost} = \sum_{i \in \text{resources}} \{C \times T\}. \quad (6)$$

6.2. Scheduling Models Performance. Cloud suppliers possess a tremendous number of servers and other handling establishments. An enormous number of virtual machines run inside a server so the resources can be utilized in the best manner. These computations observe the tasks and their needs and attend to them effectively. Optimization scheduling techniques were contrasted like round robin (RR), particle swarm optimization (PSO), first come first serve (FCFS), genetic simulated annealing (GASA), and shortest job first (SJF). To guarantee consistency, the models executed in this work utilized a comparable proportion of tasks with various lengths. The execution of various scheduling computations was finished by using IoMT-cloud tasks. Additionally, when we played out the proposed HGA, the model beats other models concerning the QoS. Also, because of the separation in the technical process, the outcomes were gainful for each model. The authors added one more parameter in contrast to their previous work.

Based on the result, it can be said that the throughput with the HGA is ideal. Table 4 shows the relationship between every one of the models against the embraced parameters. FCFS incorporates little execution time, little fulfillment time, and little holding up time as short cycles hang tight for more expanded ranges. GASA and HGA scheduling give time-sharing limits. With medium holding up time, for more modest cycles, it is not recommended where fragile traffic is incorporated. SJF is sensible for basically a wide range of circumstances. These outcomes approve our proposed methodology towards getting a proficient model. As in the exploration, it shows that the FCFS is one of the quickest with regard to the execution for the traditional model; however, this standard oddball the waiting time, with this it can prompt terminations of assignments because of the period the patients need to pause. The other AI techniques were separable and indispensable, and they certainly achieved great results, but the best is still being our proposed model. The proposed parameters are truly outstanding in defending how the models will be performed. The planning is executed with the goal that it stops after a time period is achieved. Hence, the proposed model addresses the issue by giving the best waiting time and execution time. In the approaching passage, we will examine the outcomes further with a more pictorial view. Subsequently, we can close by saying our best model being the HGA has an effective QoS.

6.3. Experimental Result Discussion and Comparison. Figure 6 displays a connection of all utilized techniques against the TFT, TWT, and the TET, and these are some of the used validation criteria's to legitimize how effective the proposed technique is. These are profoundly considered while planning to achieve a higher QoS. The outcomes demonstrate that we can achieve the most extreme utilization of

TABLE 3: Price unit for each resource.

Number of nodes	2	4	5	3	7	6	8	1
Price unit for each operation	0.8	0.7	0.2	0.9	0.6	0.4	0.4	0.2

assets. In RR, every task gets an identical instance of time, but there are certain circumstances where a typical waiting time may be a problem as depicted in the results. The outcome was examined utilizing similar information to look at the presentation of the calculation. The traditional model has the most waiting time after streamlining, despite their advantage of speed, while the other compared AI models were also efficient but not to the proposed model. Subsequently, our proposed HGA model beats all other techniques which are our standard. In addition, the proposed technique provides the least execution time, and this makes the task execution faster when contrasted with other techniques. Thus, the waiting time should be minima so users can dodge task terminations. This can decide the reasonableness of each task and the technique to use in the ideal opportunity for planning a scheduling process in the IoMT-cloud. The completion time of our model beats different models, conversely, with the way that different models have a higher completion time.

While Figure 7 depicts the throughput relationship between each technique, it shows that the best technique is the proposed HGA model with the best throughput. After a set of several tasks, endeavors were carried out to amplify the throughput. The throughput is certain to be one of the most significant criteria to depict the presence of a cycle for each time unit. The throughput outcome shows how effective the proposed model is. Thus, each assignment was divided in their tenth to depict the exhibition. During this process, our model outperforms other models in this event, during the split. Although the optimization techniques were linearly separable, they showed their efficiency. Regardless, the proposed model was the best. The throughput is the biggest amount of errands that can be finished per time unit; with this, we can conclude that the proposed model outflanks other models and fits this description well.

Figure 8 shows the relationship of usage of resources for the scheduling techniques. Moreover, the HGA utilizes the resources that are free in the run time and pick another request. In this way, the inactive waiting time is diminished in the proposed HGA calculation contrasting other techniques. Likewise, asset usage is improved separately. Nonetheless, when different assets can be used, then others can become ideal. The resource used is looked at under various total number of the makespan. The techniques have an increase in the resource utilized and thereby staying in a normal state. Regarding resource size, or amount of task increments, there is a normal increase in the normal waiting time. Thus, it can be reasoned that the HGA is most effective as opposed to the other contrasted techniques. From the figure, we can derive the effectiveness of various techniques as opposed to the proposed technique, the normal asset utilized by various techniques remains practically comparable, and

TABLE 4: Cumulative results of all models.

Traits	SJF	FCFS	RR	PSO	GASA	HGA
Total execution time	55.36	54.68	54.31	40.21	36.22	32.57
Throughput	0.72	0.73	0.74	1.01	0.99	1.21
Total waiting time	42.21	41.72	40.92	40.80	40.30	40.16
Total finish time	101.67	100.18	99.31	80.10	79.4	76.6
Availability	Success/40	Success/40	Success/40	Success/40	Success/40	Success/40
Cost	0.27	0.27	0.27	0.20	0.20	0.17
Resource utilization	0.42	0.42	0.4	0.61	0.63	0.69
Energy efficiency	0.60	0.62	0.55	0.35	—	0.30

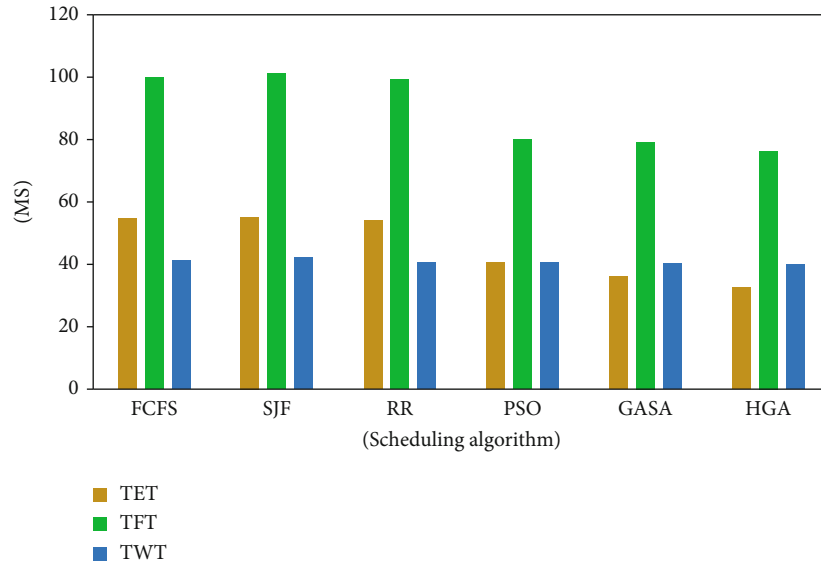


FIGURE 6: TET, TWT, and TFT of the scheduling model.

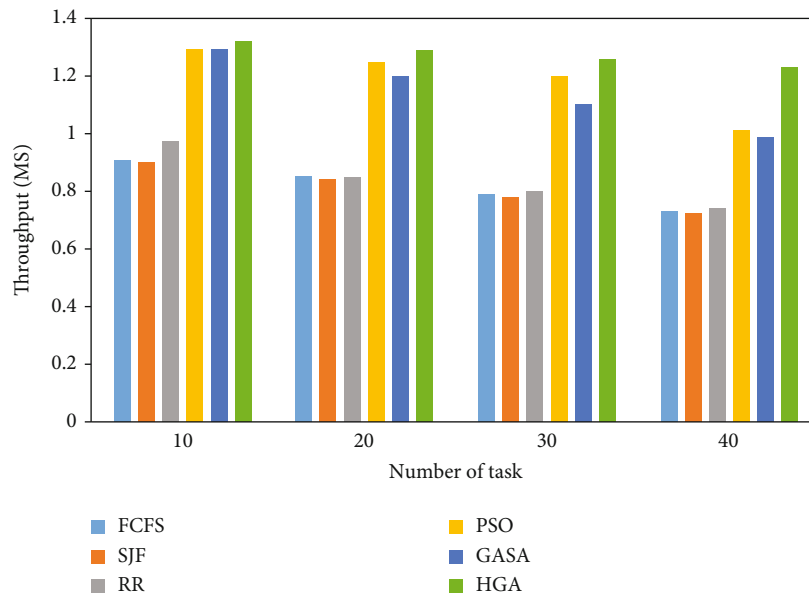


FIGURE 7: Result of the throughput.

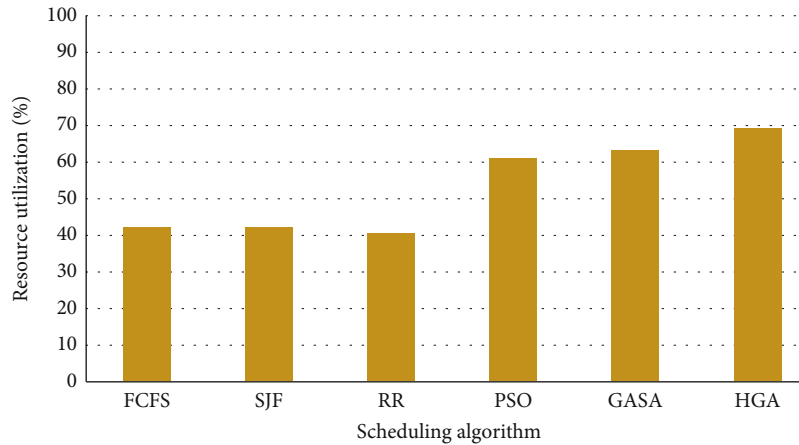


FIGURE 8: Scheduling models vs resource utilization.

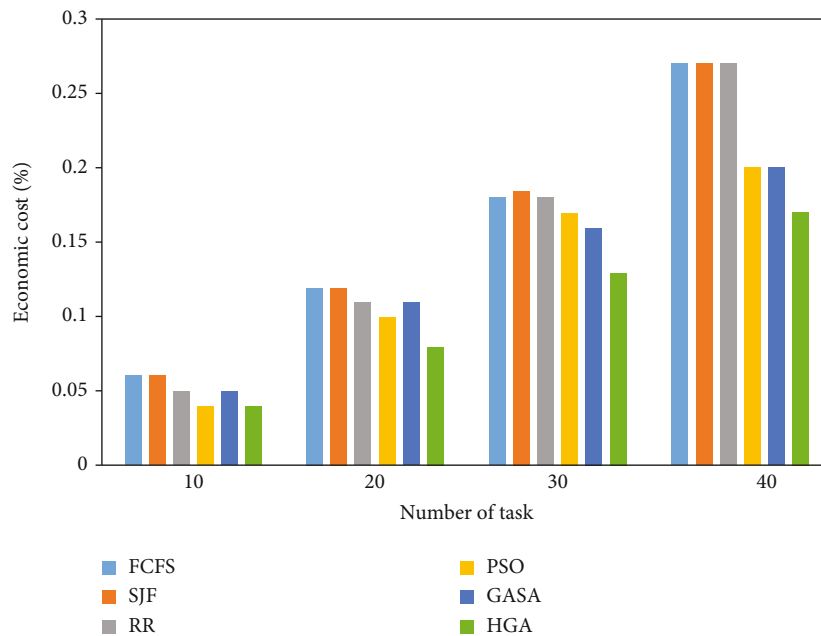


FIGURE 9: Scheduling models vs economic cost.

that signifies that it is affected by the quantity of accessible resources.

Figure 9 shows the financial expense factor. The result gotten shows the HGA model per task as a lesser cost factor. Cost depicts the impact of the charged rate over the used resource for each task. It is an evaluating factor for each center point in the IoMT-cloud environment. This impeding benefit makes it more interesting for users without the sensation of fear regarding being cheated. The HGA technique depicts a significant advantage where the rate was on a similar worth per each interaction. It is set at a level rate for each amount of resources, where making it to a reasonably higher worth will decrease the chances of the resource being picked for an undertaking. Thus, the outcome tells the best way to tackle this issue with the proposed HGA model that is to limit the expense massively. By and by, this will not suit

the client’s models as the usage in the clinical environment will need a lot of useful time and resources which will expand the expense separately. We can conclude by expressing that the HGA technique outperforms other techniques and as a base conservative expense differentiation to other contrasted models.

Figure 10 depicts the efficiency of energy consumption. Initially, the energy was calculated in KWh and later transformed to percentage to analyze its efficiency thoroughly. The parameter aims to reduce the consumption of energy. The figure shows that the HGA outperformed other models with a 30% reduced rate of energy consumption. The proposed model is compared to other metaheuristics, and it showed how efficient the model is. The PSO and GA seemed a bit fair in contrast, but the model is still lagging with regard to the efficiency of the energy consumption. This parameter,

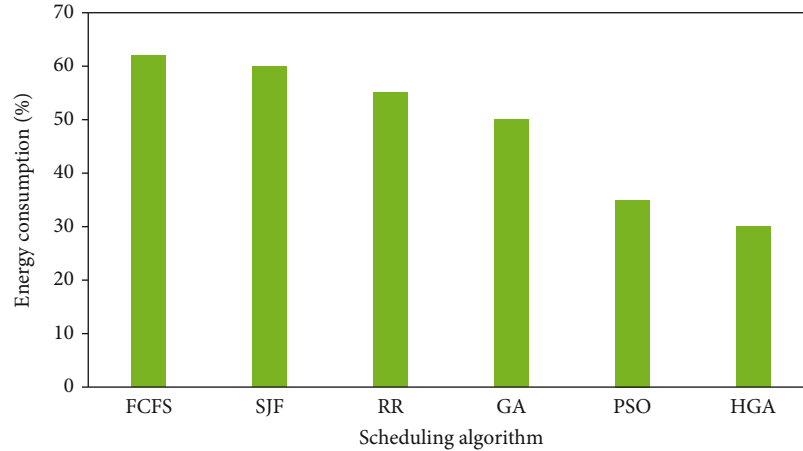


FIGURE 10: Efficiency of energy consumption comparison with HGA.

being one of the relevant parameters, will increase machine performance and at the same time will create a greener environment. The experimented known customary technique is known for its adequacy but could not outperform the proposed HGA.

Moreover, how the tested model will assume an urgent part in the clinical field where assets are utilized continuously is an eminent concern. This shows that cloud providers are expected to accomplish maxima income while thinking about QoS and solicitations from the clients. The health-care framework can be digitalized to achieve proficient association of medical care assets and administrations. With this, clinical information can be gathered, investigated, and observed. In this manner, the preliminaries show that the HGA beats different models and can be an effective method of planning for IoMT-cloud in the clinical field. Cloud clients or patients can have answer approaching solicitations without the apprehension about a task being dissolved or terminated.

7. Conclusion

The goal of this work is to style a model with a different sequency in contrast to the authors' previous work to solve task scheduling issues while at the same time-sharing resources to reach a productive QoS. The proposed work puts forth the significance of task scheduling computations and the application of AI in the IoMT-cloud environment. As we likely know, the IoMT-cloud is perhaps verifiably the most invigorating point for researchers, industry, and public zone. Thus, this theory targets developing a fast, sharp, and particular structure for task scheduling for IoMT-cloud. This evaluation put forth relies upon utilizing the present-day developments to further develop research on IoMT-cloud. This work in like manner presents an overall report between the scheduling techniques in IoMT-cloud, like the SJF, FCFS, RR, PSO, and GASA, and the proposed model being the HGA. Several parameters were used and added in contrast to the authors previous works like the TWT, TFT, TFT, resource utilization, throughput, and effi-

ciency of energy consumption and cost. This work was reauthorizing the proposed assessment with various scheduling techniques to display the ampleness of the HGA. This study gives a potential guide for clients and experts in understanding task scheduling in the cloud. From the diagrams and calculations, it was exhibited that the HGA beat other different models concerning execution time, resource utilization, throughput, and cost. The process and experiment were executed on CloudSim, which is used for showing the different scheduling process in cloud computations. The proposed HGA had an execution pace of 32.57 ms and a throughput of 1.21 ms. These two parameters are one of the most significant parameters as it satisfies both client and provider's desires against the QoS. The charts and results portray that the HGA is far better than other optimization models even with the change in sequence when veered from the cases of TWT, TET, and TFT. HGA technique can be used in IoMT-cloud as significant task response time gets reduced reasonably. Has IoMT requires high execution speed and is time-dependent. However, future examination is to be considered like completing the computation for other progression factors like speedup and stream time. In future works, it can also lessen the cost and increase the throughput with the computations to get more smoothed out results. Finally, we will update the work using several other characteristics too and will bring the outcomes as they will appear, experimenting more AI models like bee algorithm and Ant algorithm. It is acknowledged that this endeavor will help specialists and researchers whenever considered.

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The author does not have any possible conflicts of interest.

References

- [1] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [2] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.
- [3] Z. Zongbin and D. Zhongjun, "Improved GA-based task scheduling algorithm in cloud computing," *Computer Engineering and Applications*, vol. 49, no. 5, pp. 77–80, 2013.
- [4] Z. Lijuan and W. Chunying, "Cloud computing resource scheduling in mobile internet based on particle swarm optimization algorithm," *Computer Science*, vol. 42, no. 6, pp. 279–292, 2015.
- [5] X. Jie, Z. Jian-chen, and L. Ke, "Task scheduling algorithm based on dual fitness genetic annealing algorithm in cloud computing environment," *Journal of University of Electronic Science and Technology of China*, vol. 42, no. 6, pp. 900–904, 2013.
- [6] W. Fang, L. Mei'an, and D. Weijun, "Cloud computing task scheduling based on dynamically adaptive ant colony algorithm," *Journal of Computer Applications*, vol. 33, no. 11, pp. 3160–3162, 2013.
- [7] F. Al-Turjman and D. Deebak, "Privacy-aware energy-efficient framework using the internet of medical things for COVID-19," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 64–68, 2020.
- [8] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 1167–1178, 2021.
- [9] H. Chen, X. Zhu, H. Guo, J. Zhu, X. Qin, and W. Jianhong, "Towards energy-efficient scheduling for real-time tasks under uncertain cloud computing environment," *Journal of Systems and Software*, vol. 99, pp. 20–35, 2015.
- [10] R. G. Reynolds, Z. Michalewicz, and M. Cavaretta, "Using cultural algorithms for constraint handling in GENOCOP," in *Proceeding of the 4th Annual Conference on Evolutionary Programming*, pp. 298–305, MIT Press, Cambridge, MA, USA, 1995.
- [11] A. A. Hussain, O. Bouachir, F. Al-Turjman, and M. Aloqaily, "AI techniques for COVID-19," *IEEE Access*, vol. 8, pp. 128776–128795, 2020.
- [12] A. A. Hussain and F. Al-Turjman, "Resource allocation in volunteered cloud computing and battling COVID-19," in *AI-Powered IoT for COVID-19*, pp. 39–76, CRC Press, 2020.
- [13] A. Al-maamari and F. Omara, "Task scheduling using PSO algorithm in cloud computing environments," *International Journal of Grid Distribution Computing*, vol. 8, no. 5, pp. 245–256, 2015.
- [14] F. Al-Turjman, A. A. Hussain, S. Alturjman, and C. Altrjman, "Vehicle Price Classification and Prediction Using Machine Learning in the IoT Smart Manufacturing Era," *Sustainability*, vol. 14, no. 15, p. 9147, 2022.
- [15] M. Mezma, N. Melab, Y. Kessaci et al., "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497–1508, 2011.
- [16] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [17] S. Rose, "The IoT trends that no one has spoken about-read this now," 2019, <https://towardsdatascience.com/top-14-iot-trends-to-expect-in-2020-fa81a56e8653>.
- [18] Y. Kotb, I. Al Ridhawi, M. Aloqaily, T. Baker, Y. Jararweh, and H. Tawfik, "Cloud-based multi-agent cooperation for IoT devices using workflow-nets," *Journal of Grid Computing*, vol. 17, no. 4, pp. 625–650, 2019.
- [19] P. Sangwan, M. Sharma, and A. Kumar, "Improved round robin scheduling in cloud computing," *Advances in Computational Sciences and Technology*, vol. 10, pp. 639–644, 2017.
- [20] S. Oueida, Y. Kotb, M. Aloqaily, Y. Jararweh, and T. Baker, "An edge computing based smart healthcare framework for resource management," *Sensors*, vol. 18, no. 12, p. 4307, 2018.
- [21] M. Al-Khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, "Improving fog computing performance via fog-2-fog collaboration," *Future Generation Computer Systems*, vol. 100, pp. 266–280, 2019.
- [22] J. Li, T. Ma, M. Tang, W. Shen, and Y. Jin, "Improved FIFO scheduling algorithm based on fuzzy clustering in cloud computing," *Information*, vol. 8, no. 1, p. 25, 2017.
- [23] S. Elmougy, S. Sarhan, and M. Joundy, "A novel hybrid of shortest job first and round robin with dynamic variable quantum time task scheduling technique," *Journal of Cloud Computing*, vol. 6, no. 1, p. 12, 2017.
- [24] D. I. Arkhipov, W. Di, W. Tao, and A. C. Regan, "A parallel genetic algorithm framework for transportation planning and logistics management," *Access IEEE*, vol. 8, pp. 106506–106515, 2020.
- [25] Y. Gan, C. Yin, Q. Fan, and A. Li, "Improved T-matrix method for simultaneous reconstruction of dielectric and perfectly conducting scatterers," *Access IEEE*, vol. 8, pp. 143622–143631, 2020.
- [26] J.-T. Tsai, J.-C. Fang, and J.-H. Chou, "Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm," *Computers and Operations Research*, vol. 40, no. 12, pp. 3045–3055, 2013.
- [27] S. T. Maguluri and R. Srikant, "Scheduling jobs with unknown duration in clouds," *IEEE/ACM Transactions On Networking*, vol. 22, no. 6, pp. 1938–1951, 2014.
- [28] C. Cheng, J. Li, and Y. Wang, "An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing," *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 28–39, 2015.
- [29] W. Lin, C. Liang, J. Z. Wang, and R. Buyya, "Bandwidth-aware divisible task scheduling for cloud computing," *Software: Practice and Experience*, vol. 44, no. 2, pp. 163–174, 2014.
- [30] D. Ergu, G. Kou, Y. Peng, Y. Shi, and Y. Shi, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment," *The Journal of Supercomputing*, vol. 64, no. 3, pp. 835–848, 2013.
- [31] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin, and X. Liu, "Real-time tasks oriented energy-aware scheduling in virtualized clouds," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 168–180, 2014.

- [32] X. Liu, Y. Zha, Q. Yin, Y. Peng, and L. Qin, "Scheduling parallel jobs with tentative runs and consolidation in the cloud," *Journal of Systems and Software*, vol. 104, pp. 141–151, 2015.
- [33] G. Shamsollah and M. Othman, "Priority based job scheduling algorithm in cloud computing," *Procedia Engineering*, vol. 50, pp. 778–785, 2012.
- [34] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 222–235, 2014.
- [35] M. Polverini, A. Cianfrani, S. Ren, and A. V. Vasilakos, "Thermal aware scheduling of batch jobs in geographically distributed data centers," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 71–84, 2014.
- [36] A. E. Keshk, A. B. El-Sisi, and M. A. Tawfeek, "Cloud task scheduling for load balancing based on intelligent strategy," *International Journal of Intelligent Systems & Applications*, vol. 6, no. 5, p. 25, 2014.
- [37] S. Ghanbari, M. Othman, W. J. Leong, and M. R. A. Bakar, "Multi-criteria-based algorithm for scheduling divisible load," in *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*, pp. 547–554, Singapore, 2014.
- [38] H. Goudarzi, M. Ghasemazar, and M. Pedram, "Sla-based optimization of power and migration cost in cloud computing," in *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 172–179, Ottawa, ON, Canada, 2012.
- [39] F. Al-Turjman and S. Alturjman, "5G/IoT-enabled UAVs for multimedia delivery in industry-oriented applications," *Multimedia Tools and Applications*, vol. 79, no. 13-14, pp. 8627–8648, 2020.
- [40] S. A. Alabady, F. Al-Turjman, and S. Din, "A novel security model for cooperative virtual networks in the IoT era," *International Journal of Parallel Programming*, vol. 48, no. 2, pp. 280–295, 2020.
- [41] F. Al-Turjma, "Intelligence and security in big 5G-oriented IoNT: an overview," *Future Generation Computer Systems*, Volume, vol. 102, pp. 357–368, 2020.
- [42] S. Ghanbari, M. Othman, M. R. A. Bakar, and W. J. Leong, "Priority-based divisible load scheduling using analytical hierarchy process," *Applied Mathematics & Information Sciences*, vol. 9, no. 5, pp. 25–41, 2015.
- [43] B. Radojevic and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments," in *2011 Proceedings of the 34th International Convention MIPRO*, pp. 416–420, Opatija, Croatia, 2011.
- [44] S. Ghanbari, M. Othman, M. R. A. Bakar, and W. J. Leong, "Multi-objective method for divisible load scheduling in multi-level tree network," *Future Generation Computer Systems*, vol. 54, pp. 132–143, 2016.
- [45] S. Goswami and A. Das, "Optimization of workload scheduling in computational grid," in *Proceedings of the 5th international conference on Frontiers in intelligent computing: theory and applications*, pp. 417–424, Odisha, India, 2017.
- [46] K. Kaur and A. Kaur, "A hybrid approach of load balancing through VMs using ACO, MinMax and genetic algorithm," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 615–620, Dehradun, India, 2016.
- [47] M. S. Pilavare and A. Desai, "A novel approach towards improving performance of load balancing using genetic algorithm in cloud computing," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–4, Coimbatore, India, 2015.
- [48] R. R. Patel, S. J. Patel, D. S. Patel, and T. T. Desai, "Improved GA using population reduction for load balancing in cloud computing," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2372–2374, Jaipur, India, 2016.
- [49] A. A. S. Farrag, S. A. Mahmoud, and E. S. M. El-Horbaty, "Intelligent cloud algorithms for load balancing problems: A survey," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 210–216, Cairo, Egypt, 2015.
- [50] H. A. Makasarwala and P. Hazari, "Using genetic algorithm for load balancing in cloud computing," in *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–6, Ploiesti, Romania, 2016.
- [51] K. V. Kavitha and V. V. Suthan, "Dynamic load balancing in cloud-based multimedia system with genetic algorithm," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, pp. 1–4, Coimbatore, India, 2016.
- [52] S. Dam, G. Mandal, K. Dasgupta, and P. Dutta, "Genetic algorithm and gravitational emulation-based hybrid load balancing strategy in cloud computing," in *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pp. 1–7, Hooghly, India, 2015.
- [53] L. Hong and G. Yufei, "GACA-VMP: virtual machine placement scheduling in cloud computing based on genetic ant Colony algorithm approach," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pp. 1008–1015, Beijing, China, 2015.
- [54] A. A. Hussain and F. Al-Turjman, "Artificial intelligence and blockchain: a review," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 9, 2021.
- [55] A. A. Hussain, B. A. Dawood, and F. Al-Turjman, "Application of AI techniques for COVID-19 in IoT and big data era: a survey," in *Artificial intelligence and machine learning for COVID-19*, pp. 175–211, Springer, 2021.
- [56] A. A. Hussain, F. Al-Turjman, and M. Sah, "Semantic web and business intelligence in big-data and cloud computing era," in *The Proceedings of the Third International Conference on Smart City Applications*, pp. 1418–1432, Turkey, 2021.
- [57] A. A. Hussain, F. Al-Turjman, E. Gemikonakli, and Y. K. Ever, "Design of a navigation system for the blind/visually impaired," in *International Conference on Forthcoming Networks and Sustainability in the IoT Era*, pp. 25–45, Cyberspace, 2021.
- [58] A. A. Hussain and K. Dimililer, "Student grade prediction using machine learning in Iot era," in *International Conference on Forthcoming Networks and Sustainability in the IoT Era*, pp. 65–81, Cyberspace, 2021.
- [59] A. A. Hussain, B. A. Dawood, and F. Al-Turjman, "IoT and AI for COVID-19 in scalable smart cities," in *International Summit Smart City 360°*, pp. 3–19, Springer, 2021.