Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.022783

ARTICLE

IoMT-Cloud Task Scheduling Using AI

Adedoyin A. Hussain[1,2,*] **and Fadi Al-Turjman**[3,4]

[1]Computer Engineering Department and Research Centre for AI and IoT, Near East University, Nicosia, Turkey

[2]Computer Engineering Department, Cyprus West University, Gazimagusa, Turkey

[3]Artificial Intelligence Engineering Department and AI and Robotics Institutes, Near East University, Nicosia, Turkey

[4]Research Centre for AI and IoT, Faculty of Engineering, University of Kyrenia, Kyrenia, Turkey

*Corresponding Author: Adedoyin A. Hussain. Email: hussaindoyin@gmail.com

Received: 25 March 2022 Accepted: 13 June 2022

ABSTRACT

The internet of medical things (IoMT) empowers patients to get adaptable, and virtualized gear over the internet. Task scheduling is the most fundamental problem in the IoMT-cloud since cloud execution commonly relies on it. Thus, a proposition is being made for a distinct scheduling technique to suitably meet these solicitations. To manage the scheduling issue, an artificial intelligence (AI) method known as a hybrid genetic algorithm (HGA) is proposed. The proposed AI method will be justified by contrasting it with other traditional optimization and AI scheduling approaches. The CloudSim is utilized to quantify its effect on various parameters like time, resource utilization, cost, and throughput. The proposed AI technique enhanced the viability of task scheduling with a better execution rate of 32.47 ms and a reduced time of 40.16 ms. Thus, the experimented outcomes show that the HGA reduces cost as well as time profoundly.

KEYWORDS

Artificial intelligence; IoMT; hybrid genetic algorithm; cloud

1 Introduction

Scheduling tasks is classed as one of the focal problems for computing in IoMT-cloud. IoMT-cloud has progressed with the improvement of PC and gadget advancement. IoMT is the connection of medical devices where these devices can communicate with each other and share resources for medical assistance. A reasonable scheduling technique is required to schedule these IoMT requests to cloud resources. Task scheduling for the cloud is one of the main advances in the IoMT stage, which impacts the whole execution of the cloud resource. This prompts the execution of all tasks efficiently and also provides patients with formidable QoS [1,2]. Numerous investigations show that the IoMT-cloud task scheduling problem is termed an NP-hard problem, which has been concentrated by various analysts. Zhu et al. [3] proposed a scheduling computation for tending to the cloud task to further develop scheduling estimation, which can get a more unobtrusive time and lower cost for each process. It has achieved extraordinary results in the field of arranging resources to cloud tasks ensuing in finishing a huge number of coherent tasks. Zhou et al. [4] proposed particle swarm optimization (PSO) to deal with the idea of the organization of users. Xu et al. [5] prepared a hereditary reenacted tempering estimation for task arrangement with twofold fitness, this can effectively change the solicitations of the clients for the properties of tasks and work on the client's satisfaction appropriately. It incorporates a moderate speed of processing but it essentially caught more waiting time. Wang et al. [6] used the procedure for tending to the cloud task scheduling by exceptional self-changing ant colony optimization (ACO) in handling the scheduling of tasks.

To work on task scheduling adequacy in the IoMT-cloud stage, the cloud environment has to be viewed and studied. Fig. 1 provides a proper view of the IoMT-cloud stage. Cloud is a proficient headway for computation. It encompasses data storage, management, and manipulation in large volumes, and uses that data to control the transformative cycle of principle in people space [7–11]. This reduces the outright period of manpower and lessens the cost of the health system. This is a foremost advancement that uses the possibility of business execution of computer programming with patients publicly [12]. It relies upon splitting resources between clients utilizing the virtualization methodology. IoMT-cloud is one more advancement gotten from grid computing and it insinuates using enlisting resources in an organization and providing for beneficiaries on interest through the Internet [13]. Scheduling in the cloud is one of the major factors in IoMT. It is considered to be the essential factor that controls other execution models, for instance, openness, flexibility, patient resource sharing, and power use. Regardless, there are various troubles normal in IoMT computing. A high execution rate can be given by the scheduling technique, 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.

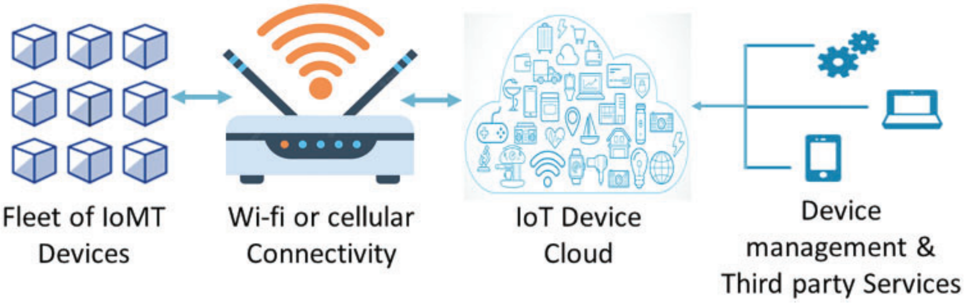
Fleet of IoMT Devices
Wi-fi or cellular Connectivity
IoT Device Cloud
Device management & Third party Services

Figure 1: IoMT-cloud platform illustration

The ideal critical process of IoMT experimentation is that it propels the authentic use of resources [14]. Each impacts the other. Fitting these IoMT tasks 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 under 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 and patients [15]. The 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 one of the huge central purposes of the IoMT cloud computation. Nonetheless, the organization may be responsible for paying additional costs for this proposition. An example of the IoMT-cloud trends is depicted in Fig. 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 use depends on the scheduling and resource weight, rather than the unpredictable designation of resources. Task scheduling in IoMT-cloud is for the most part used for handling complex endeavors (client requests). Such arranging computations impact the resources.

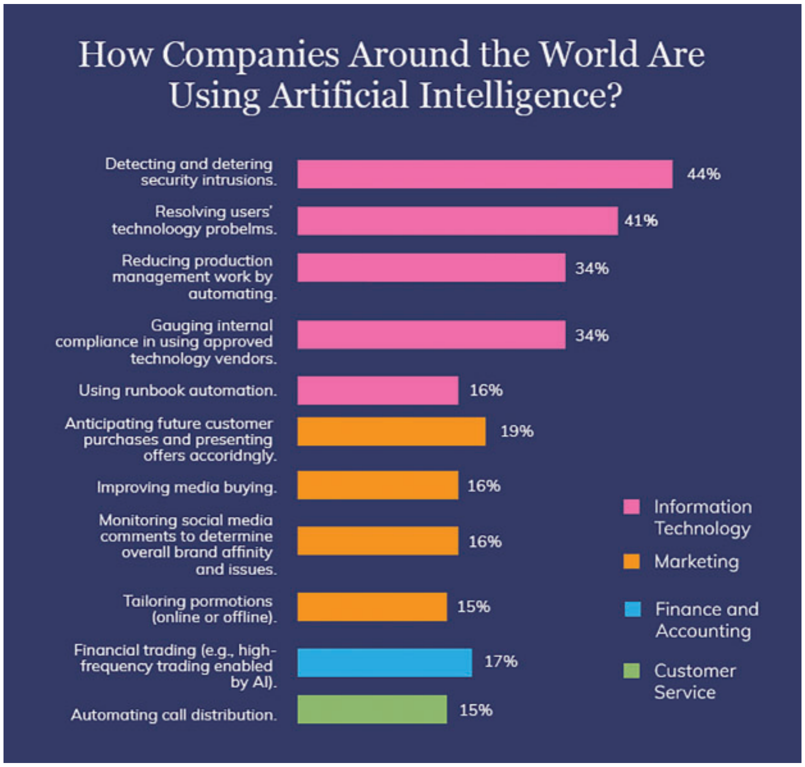

Figure 2: Trends of IoMT-cloud [17]

In this work, the contribution is provided below:

- Describing major factors that are required for task scheduling in IoMT-cloud.
- Task scheduling and optimization survey in IoMT-cloud.
- A portrayal and analysis of the result gotten from the examination.
- Proposing an AI technique as a Hybrid for IoMT-cloud scheduling issue.
- Summing up 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 the experiment is 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.

Table 1: Abbreviations used in the work

Terms	Meaning
SJF	Shortest Job First
ICT	Information and Communication Technologies
SaaS	Software as a Service
RR	Round Robin
IaaS	Infrastructure as a Service
TET	Total Execution Time
AI	Artificial Intelligence
ML	Machine Learning
TFT	Total Finish Time
TWT	Total Waiting Time
NP	Nondeterministic Polynomial
GA	Genetic Algorithm
ACO	Ant Colony Optimization
IoMT	Internet of Medical Things
PSO	Particle Swarm Optimization
PaaS	Platform as a Service
QoS	Quality of Service
FCFS	First Come First Serve
PC	Personal Computer

2 Background

This section discusses the background of the proposed work. The traditional dynamic and static optimization approaches in IoMT are discussed. In addition, the AI approach in IoMT is also discussed.

2.1 Traditional Optimization Techniques

Traditional optimization procedures are fundamental when it comes to scheduling in the IoMT cloud. The IoMT assets are sensitive concerning time and require a quick execution. The requirement for a decent scheduling approach is imminent. There are different conventional strategies like Round Robin (RR), First Come First Serve (FCFS), and Shortest Job First (SJF) for optimization in the IoMT cloud stage. These conventional procedures birth other methods since they are straightforward, fast, and deterministic and the arrangements are precise [18]. Several studies have been completed to work on the execution of these conventional methods [19,20]. The conventional optimization methods can additionally be partitioned into static and dynamic. The process lined in light of need is examples of the static scheduling procedure while the contrary will be dynamic [21]. The latter usually considers a powerful factor like applying task quantum time for task fairness.

2.1.1 First Come First Serve (FCFS) in IoMT-Cloud

This is a customary methodology, any 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 [22]. It depends upon the standard of FIFO with less complexity than other computations techniques [23]. Separately each cycle is taken from the head of the line. This process is immediate and expedient. Whenever we have immense requests, all requests delay until the primary occupation is done. To evaluate the achievement of 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 a static methodology. The FCFS has these qualities:

- Prioritization depends on the main request and this makes each cycle toward the end finish before some other added cycle.
- 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.

2.1.2 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 [24]. 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 [25]. Be that as it may, it has a more drawn-out finish time. This procedure is said to be a static scheduling procedure. This is a direct result of tasks with high processes being left unattended while little processes are taken care of. It has these processive traits:

- It will always be aware of the next task process.
- It lessens the waiting time for the task process as it processes little tasks before huge ones.

2.1.3 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 from the static type as a result of its dynamicity [26]. Right away, all task processes are given equivalent time for execution which is once in a while called quantum time. All processes are kept in the solicitation as they show up [27]. 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:

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

2.2 AI Optimization Approach

Artificial intelligence application in the IoMT-cloud is the merging of the AI capacities of intellectual man-made prowess with cloud-based systems. The IoMT assets are time delicate and require a quick execution, the requirement for a decent scheduling technique is needed. This requirement for AI is prominent due to its capacity and attributes. The use of AI clears a path for more viability, flexibility, and key comprehension than the world has seen up to this point [28]. For instance, utilizing ML models to recognize cancers, countless radiology reports are used to set up this structure [29]. This will allow the technological scheduling structure to run routine tasks together and productively.

2.2.1 Genetic Algorithm (GA) in IoMT-Cloud

The GA is an AI strategy that has gained ascendance 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 [30]. The general population is taken self-assertively to fill in as the early phase of 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 [31]. The study here 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 beneath:

- Initialization: Generates an initial population.
- Fitness: Based on the fitness value, calculate for each chromosome.
- Mating pool: Select the 2 best chromosomes after wellness handling; 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.
- Fitness: Based on the fitness value, calculate for each chromosome.
- Repeat 2 to 5 until meeting the end condition. A stopping condition may be the number of cycles.
- 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 overcome the issue of scheduling and resource assignment. Tsai et al. [32] put forth a multi-object technique that uses better differential progression computation. Regardless, assortments in the tasks are not considered in this philosophy. In any case, further upgrades can even presently be made. However, this current technique gives a cost and time model for conveyed scheduling. This approach does not show the genuine utilization of resources. Maguluri et al. [33] put forth a heap changing and arranging estimation that does not consider work sizes. A programming nonlinear model has been formed to disperse resources for tasks. Cheng et al. [34] introduced the preparation of tasks reliant upon an excursion lining model. They have illustrated the association between task and energy assurance by resource apportioning. Nonetheless, Lin et al. [35] proposed the scheduling of tasks while pondering information transmission as a resource. Ergu et al. [36] proposed Analytic Hierarchy Process (AHP) situating-based endeavor arrangement. The proposed system doesn't focus on rashly finishing the processes and starvation. Zhu et al. [37] proposed an acquainted moving skyline approach with planned tasks. They considered the FCFS method for managing demands when resources are available. Subsequently, Liu et al. [38] proposed anticipating equivalent extraordinary weights based on incoming demands. Shamsollah et al. [39] put forth a need-based business scheduling estimation for use in disseminated registering. Rodriguez et al. [40] put forth the use of a meta-heuristic improvement to diminish execution costs through task arranging. Multi estimates decisions and various credits are considered. Polverini et al. [41] 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, Keshk et al. [42] proposed the usage of modified bug area upgrade in load changing. Ghanbari et al. [43] proposed a system subject to a multi-guidelines computation for arranging specialist load. This strategy works on the makespan of a work. Thus, Goudarzi et al. [44] put forth a resource assignment problem that means restricting the full-scale energy cost of appropriated scheduling structures while meeting the foreordained client-level SLAs according to a probabilistic point of view [45–47]. 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, Ghanbari et al. [48] 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 Radojevic et al. [49] introduced a central weight changing the decision model in the cloud. They looked at using heuristic approaches like MAX-MIN. A couple of makers have proposed a heuristic estimation to handle task arranging and resource task issues portrayed already. Regardless, this model is lacking in choosing the capacities of center points and arrangement nuances. The complete structure has no support, as needs are achieving a lone reason for dissatisfaction. Moreover, Ghanbari et al. [50] and Goswami et al. [51] focused on arranging endeavors while contemplating various goals. This bleeding-edge convinces the makers of this assessment to coordinate additional examination on task planning and resource distribution.

Another scheduling approach is using the Ant Colony Optimization (ACO) planning computation. Kaur et al. [52] proposed a reasonable weight-changing computation by using a hybridization of bug region movement strategy, 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 strong method to restrict the cost of movement of VM and keep up the SLA (Service Level Agreement) which is a QoS factor. As GA aimlessly picks the processors and a short time later applies the inherited estimation, the fittest processors track down the open door, and the VM which has lower need starves. Through this, the need is apportioned to VM to extend the response period of the system and to achieve better weight change. Pilavare et al. [53] thought about the procedure that deals with the starvation issue in work change. To vanquish this trouble, they used innate computation with the logarithmic least square strategy. Thusly, it helps with growing the response time which prompts better execution of the structure and cares for consistency. With this, Patel et al. [54] put forth an Improved GA by using the fragmented people decline procedure Planned Parenthood of the Rocky Mountains (PPRM). Farrag et al. [55] have audited keen cloud estimations to change the load and proposed AntLion Optimizer (ALO) to give better results in changing the store in the cloud. After this cycle, GA is applied to the new population and observed in health regard. This gives huge courses of action. While ALO handles the gigantic issue in space [56–58].

Fundamental GA has three principle exercises. Fundamental GA has terms called mutation, crossover, and fitness work [59,60]. Makasarwala et al. [61] 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. Kavitha et al. [62] have proposed a Cloud-based approach for the most part 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 requests and ensures better QoS, high interoperability, and flexibility. Dam et al. [63] have proposed a cross variety of genetic computations like Genetic Ant Colony Algorithm-Virtual Machine Placement (GA-GEL) estimation for the VM load balance process in the cloud. Liang et al. [64] 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 [65–70]. Yet as discussed in the literature, there are still various areas that need tending to, this work proposed here is aimed to settle these issues. The connection of the literature with our work is depicted in Table 2.

Table 2: Model comparison with others

Scheduling algorithm	Performance	Execution time	Scalability	Response time	Quality of service	Cost
Adaptive energy-efficient task scheduling algorithm	✓					
Multi-objective task scheduling					✓	✓
Green energy-efficient based task scheduling algorithm					✓	
Load balancing task scheduling algorithm	✓		✓			
Particle swarm optimization based task scheduling			✓		✓	
Our research	✓	✓	✓	✓	✓	✓

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 has, P_m, and T_n, respectively. Let's say the available P_m processors for some set of tasks T_n, with no sharing during execution. Let ECT is the expected completion time, which contains the 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 resource utilization and minimize time, the tasks have to be efficiently scheduled or mapped appropriately on the resources available. The depiction is shown in Fig. 3.

$T_{n\,=}$ $(T_{1,}\ T_{2\,\ldots}\ T_n)$,

$P_{m\,=}$ $(P_{1,}\ P_{2\,\ldots}\ P_m)$

where T_n is the set of tasks and P_m is the set of resources.

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

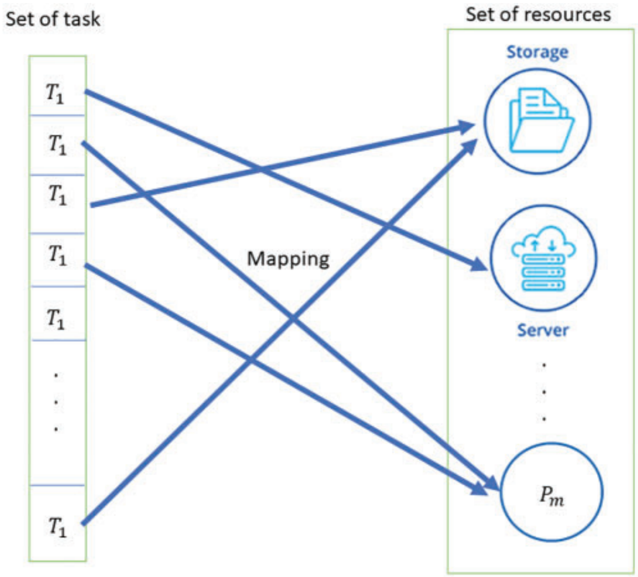

Figure 3: Scheduling task mapping

5 Proposed Methodology

This section discusses the method and gives more perspectives on the proposed system as a whole. The system description is portrayed in Fig. 4. It shows the participant in the scheduling on the IoT-Cloud platform. The user requests a task and he is responded to accordingly by the service provider, by providing the appropriate resource based on its scheduling algorithm.

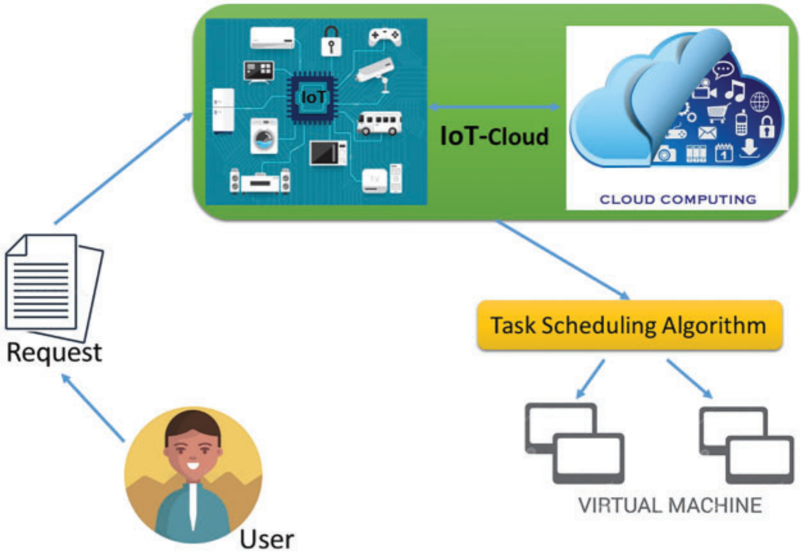

Figure 4: The system architecture

5.1 Scheduling Model for IoMT-Cloud

The IoMT-cloud has different characteristics which deal benefits to the end client. The major features of IoMT-cloud are self-redesigned, adaptability, and customization. The structure intends to work on the display of scheduling in IoMT, while simultaneously diminishing computational costs [71–74]. 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 [75]. 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 utilizing the proposed AI approach which will be the HGA. Furthermore, we separate the essentials and consequences of utilizing Quality of Service (QoS) from 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 patients request the resources, and the cloud provider is answerable for the assignment of the anticipated resource, so the client avoids the encroachment of the service level agreement (SLA). For the techniques for planning IoMT-cloud resources, the pattern of task scheduling helps to get tasks from the clients and get a solicitation from the cloud information service (CIS) for their properties and open resources. The cloud scheduler has to be skilled to appoint different virtual machines (VMs) to a different process. Thus, the scheduling structure in the IoMT-cloud is shown in Fig. 5. The proposed AI technique will use a hybrid genetic algorithm (HGA) with the blend of a dynamic round-robin and a local search, 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 desired and supplier desired) and the best outcome from these estimations is talked about in the accompanying subsection.

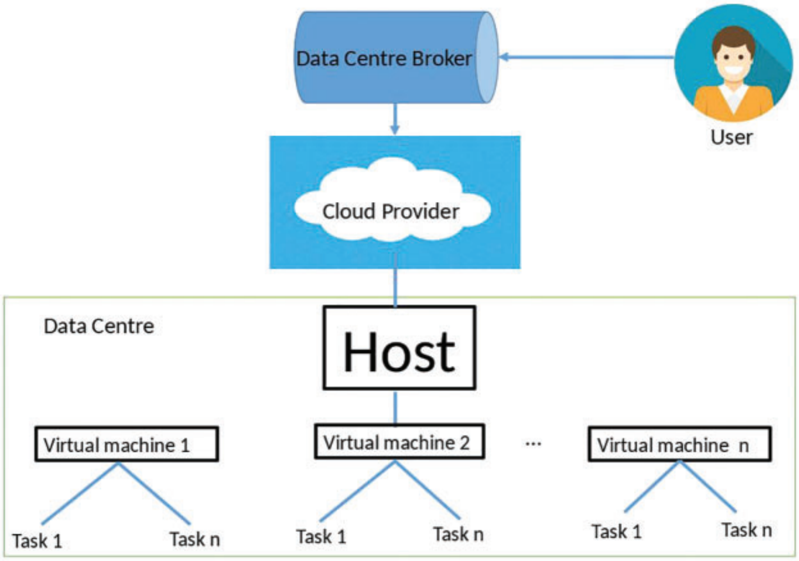

Figure 5: Task scheduling process structure

5.2 The Proposed 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 population 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 repeated until satisfactory successors are made. Here, the proposed HGA will combine a dynamic round-robin and a local search algorithm known as hill climbing. This is also known as the combination of a meta-heuristic optimization technique, a traditional optimisation technique, and a local search approach. The flowchart of the HGA in the IoMT-cloud is displayed in Fig. 6. The HGA in the IoMT-cloud process is shown as follows:

- Initializing the process:

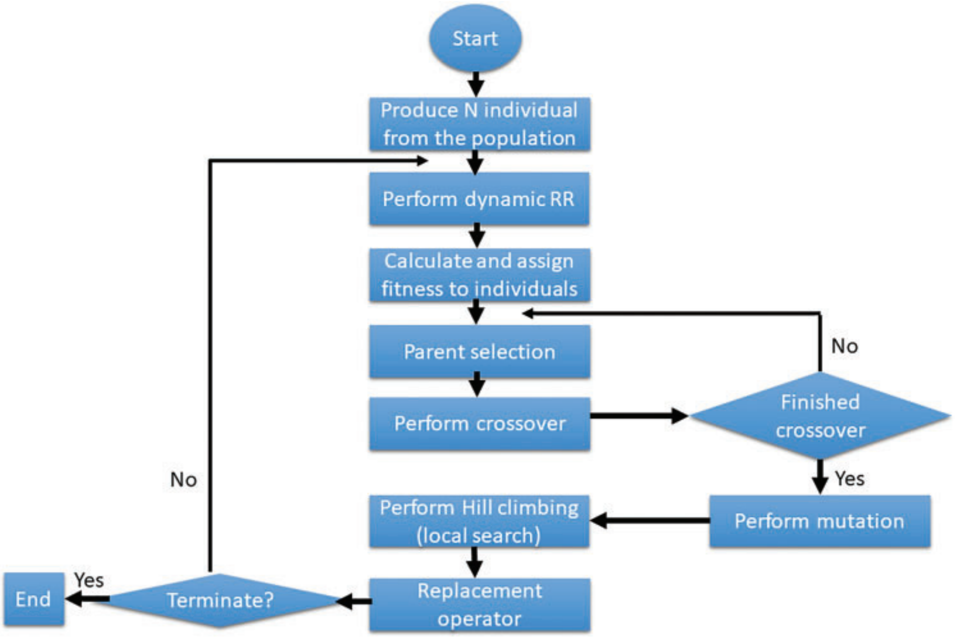

Figure 6: HGA flowchart

Introductory generation of population 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).

- 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, T8) with their respective completion times of (10, 5, 5, 5, 10), in this case, the quantum time will be given as the median of these processes. Implementing this will grant task fairness for the task with longer and minima time process. This procedure will continue till 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 Eq. (1):

$$\mathrm{TCT} = \max(CT_{n,m}) \tag{1}$$

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 using Eq. (2):

$$P_{nm} = \frac{Cn}{PSm} \tag{2}$$

After getting the processing time, the processing time of every task in the VM has to be calculated using Pj in Eq. (3):

$$P_m = \sum\nolimits_{i=1}^{n} P_{nm} \tag{3}$$

- 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, then whichever chromosome possesses a lesser completion time is the best chromosome.

- 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 chromosome 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.

- Hill climbing operation:

The newly generated offspring 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 so the elitism will not become a random loop.

- Replacement:

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

- The resulting output will be the best chromosome.
- End process.

This process comprises three fundamental processes: initial population, round-robin calculation, and finally hill climbing. These tasks are clarified beneath:

- Initial population: Each individual in the population is a possible solution for the IoMT task scheduling problem. Thus, all possible solution in the population space is encoded into paired genes consisting of task and VMs. From this, the initial value of possible solutions will be generated at random.
- Round robin: The goal of this process is to generate task fairness. The round-robin is known as a traditional optimization technique. It is fast and generates a good possible solution for scheduling problems. The pool of chromosomes goes through the dynamic round-robin, where a quantum time will be given for task fairness. From this process, the fitness value can then be calculated for a possible better solution for the population.
- Hill climbing: Hill climbing is a local search operation. While utilizing evolutionary algorithms, the speed of convergence for possible solutions is always low. To generate a better solution local search algorithms are always implemented. Thus, to optimize the elitism so it will not be regarded as a random search in the search space, the hill-climbing technique is used.

The normal GA changes the load in the IoMT-cloud by allotting tasks to the available virtual machines. Regardless, it is not effective in resource use, which infers it fails to utilize every single open virtual machine. It reliably delegates the task to just part of the VMs. The proposed model screens every one of the free virtual machines. On account of which machines stay latent while a couple of machines are over-loaded. When another task appears, it is checked that assuming a free machine is open, the task is allocated to that particular machine. Generally, in normal GA, the resources are not suitably utilized. So, this issue is taken care of by improvement with the proposed HGA. If no free virtual machine is open, by then, the endeavor is given to that machine whose current task will be done in a lesser time. Thusly, all the VMs are fittingly utilized and no VM remains idle and no VM is abused. The proposed GA will give a better yield to cost, TFT, TWT, and all the VMs are in operation.

5.3 Experimental Process

Assumptions to be viewed while planning the process in the IoMT-cloud is:

- Each task is dispensed to only a solitary VM resource.
- The task will be greater than the amount of VMs. This infers that every VM ought to process more than one task.
- The task is not obstructed once their executions start.
- The lengths of the task will be of various sizes.
- 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.
- VMs are independent concerning resources and control.

The IoMT-cloud scheduling process will be of three main levels. Fig. 7 shows the process and the discussion is below.

- The first (Task): This is various tasks or requests sent from the clients or the patients for execution.
- Second (scheduling): This consists of various procedures or rules that are used for planning or scheduling these requests to the appropriate resource. The main process is to gain more resource utilization while at the same time reducing the timing factors. The timing factor consists of the total time for each process.
- Third (VM): This is the virtualization of resources provided by the cloud provider to the client for their convenient use to execute their requests.

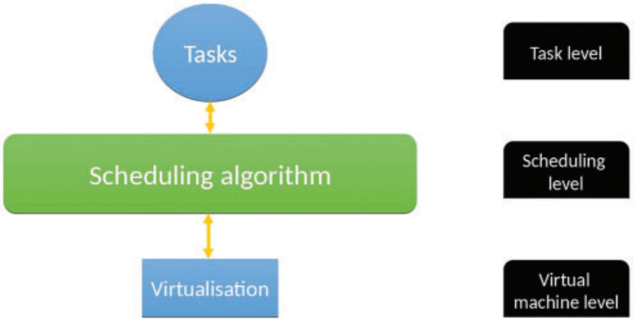

Figure 7: Task scheduling experimental system phase

5.4 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 stacking information into the application, information depiction, and construction attestation, showing the outcome, an illustration of depiction, and finally, looking at the information. The yield will be depicted visually in this work for more understanding.

5.5 Computational Environment

Eclipse is an environment for data evaluation and authentic approaches. The assessments were executed utilizing this IDE. It is open-source software that implements the use of AI methodologies. Cloudsim is used for simulating in the IoMT-cloud stage. Java maybe the most outstanding programming language, and it offers different libraries that can oversee information science attempts, for example, information assessment, information pre-dealing, and explicitly, working of different techniques. It is correspondingly the most trademark and experienced language and it was utilized in this evaluation. The research is implemented using a pc with, intel i7 Processors: 2.3 Ghz, GPU: EFORCE, Disk: 1 TB, RAM: 12 GB.

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 with various optimization and hybrid approaches. The models have been endeavored with various settings to accomplish the most fundamental TWT, TET, TFT, cost, 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 were 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. In Table 3, we depict a segment of what the IoMT task length looks like.

Table 3: Illustrating various task length

Task	Length (ms)
T1	100000
T2	70000
T3	10000
T4	5000
T5	15000
T6	90000
T7	150000
T8	200000
T9	25000
T10	60000

6.1 Parameters and Metrics

For validating the results of our proposed techniques with other models, the computational metrics below are used for this work. Though, Table 4 shows the utilized parameters.

Table 4: Used parameter

Parameter	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

TET (Total execution time): This is the total amount of time spent by each process from its arrival till it finishes. It insinuates the time between the snapshot of accommodation of a task and the hour of its perfection. Thusly what measure of time it expects to execute a cycle is in like manner a critical element.

TWT (Total waiting time): It is the time an assignment sits tight for execution when a couple of tasks are battling for the resource. It is the outright time spent by the interaction or occupation in the pre-arranged state ready to be executed.

TFT (Total completion time): It is the distance in time that breaches from the start of a task till it wraps up. It is the time at which a task or a cycle completes its execution.

Resource utilization: This boundary is one of the principal importance in task scheduling. It is one more boundary that shows the amplification of the use of assets. Though, providers need to achieve maxima gains by delivering a restricted measure of assets. This means assets will be kept occupied. Also, throughput and reaction time is critical, however, one more boundary for execution measurements for the cloud is the utilization of assets. The equation underneath shows how it is determined, where n is the number of assets and i completions time for every asset using Eq. (4):

$$\text{Average resource utilization} = \frac{\text{Time is taken by resource i to finish all the task}}{\text{Makespan}} \times n \tag{4}$$

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. The accessibility status is a triumph when the right resource is being consigned to the VM. Resource availability is one of the principal parts of scheduling.

Throughput: The scheduling approach should want to grow the number of assignments dealt with per time unit. The throughput is the proportion of work completed in a unit of time. It might be described as the number of cycles executed by the VM in a given proportion of the time. Throughput is a way to deal with finding the capability of the scheduling approach.

Cost: This financial expense will be founded on the amount of time spent by the client on a specific asset. This shows the financial expense which portrays the aggregate sum that should be paid by the client to the organization for the asset being used. The Eq. (5) shows how it is determined where T implies the time the asset is being used and C hints at the financial expense of the asset per unit time. Table 5 below portrays the value factor in the unit.

$$Cost = \sum\nolimits_{i \in resources} \{C \times T\} \tag{5}$$

Table 5: 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

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. Traditional optimization techniques were contrasted like first come first serve (FCFS), round-robin

(RR), and shortest job first (SJF). Also, other AI models were contrasted like particle swarm optimization (PSO) and Genetic simulated annealing (GASA). 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.

Based on the result, it can be said that throughput with the HGA is ideal. Table 6 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 uptime, 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 for getting a proficient model. As in the exploration, it shows that the FCFS is one of the quickest executions for the traditional model. However, this standard oddballs the waiting time, which can prompt terminations of assignments because of the period the patients need to pause. The other AI techniques were separable and indispensable, they certainly achieved great results but the best 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 HGA model has an effective QoS.

Table 6: 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	32.47
Throughput	0.72	0.73	0.74	1.01	0.99	1.23
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
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
Resource utilisation	0.42	0.42	0.4	0.61	0.63	0.69

6.3 Experimental Discussion, Result, and Comparison

Fig. 8 displays the connection of all utilized models against the TFT, TWT, and the TET, these are some of the used parameters to legitimize how effective our model is. The timing factor is considered one of the prominent factors regarding IoMT scheduling. These are profoundly considered while planning to achieve a higher QoS. The outcomes demonstrate that we can achieve the most extreme utilization of assets. In RR, every task gets an identical proportion of time, but there are a couple of circumstances where typical waiting time can be a problem as displayed in the outcomes. 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 its 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 model delivers the least execution time, this makes the execution of errands quicker contrasted with different models. This waiting time should be minimal so users can dodge task terminations. With this, we can decide the reasonableness of the task and which 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.

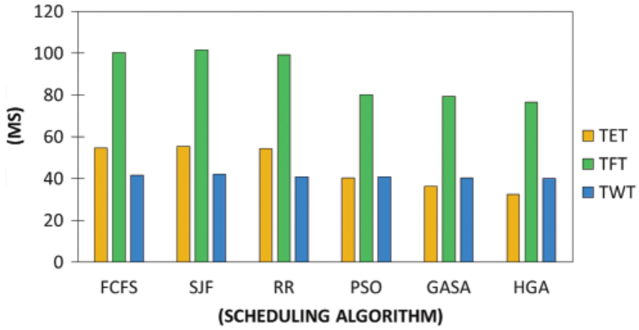

Figure 8: TET, TWT, and TFT of the scheduling model

While Fig. 9 shows the relationship against the throughput, it shows the best technique with the best throughput is the proposed HGA model. This is a prominent factor for the service provider as the amount of throughput determines how efficient the QoS is. After a series of several tasks, endeavors were made to amplify the throughput. The throughput is one of the most mind-blowing legitimizing boundaries to show the presence of a cycle for each unit time. This outcome portrays how effective our model is. Thus, each assignment was parted into its tenth to show the exhibition. During this process, our model outperforms other models in this event, during the split. Although the traditional and other AI optimization models were linearly separable and they showed their efficiency individually. Regardless, the proposed model was the best. The throughput is the biggest number of errands that can be finished per unit time, with this, we can conclude the proposed model outflanks other models and fits this description well.

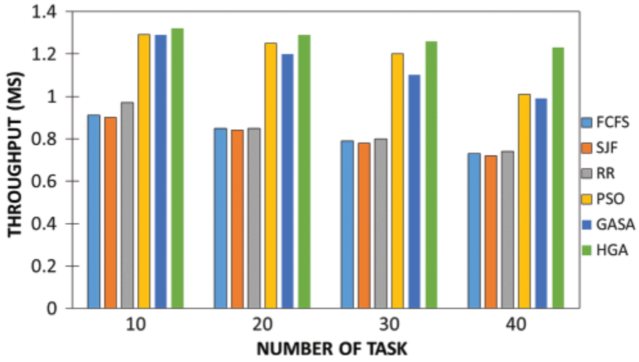

Figure 9: Result of the throughput

Fig. 10 shows the relationship of usage of resources for the scheduling model. Moreover, the proposed HGA uses the assets that are free during the run time and picks another task. The number of resources used also helps in executing patients' requests faster. In this way, the inactive waiting time is diminished in the proposed HGA calculation contrasting other different models. Likewise, asset usage is improved separately. Nonetheless, when different assets can be used, then others can become ideal. The asset used is looked at under different total counts of the makespan. The traditional and other AI models have an increment in the asset used and afterward stay in a consistent state. As the size of the resource, or how much the task increments, there is an ordinary ascent in the normal waiting time. In this manner, we can reason that the HGA is the most effective as opposed to the other contrasted models. From the figure, we can likewise derive the effectiveness of different models as opposed to our model, the normal asset used by different models remains practically comparable, and that implies it is impacted by the number of accessible assets.

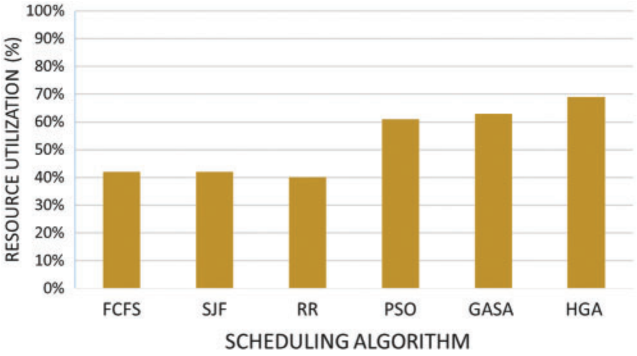

Figure 10: Scheduling models *vs.* resource utilization

Fig. 11 shows the financial expense factor. The cost is to see the effect of the charged worth rate over the pre-owned methodology for information conveyance. The outcome gotten shows the HGA model per task as a lesser expense factor. This criterion balances between the user and the provider side, as the provider desires to gain from the utilized resources. It is an assessing factor for each center point in the IoMT cloud pack. This obstructing advantage makes it more intriguing for clients without the feeling of dread toward being cheated. The HGA model shows a promising benefit where the rate was on a comparative worth for each process. This was set as a level rate for each number of resources, where setting it to a reasonably high worth would decrease the chances of the resource being picked for an undertaking. The outcome tells the best way to tackle this issue with the proposed HGA model is to limit the expense massively. By and by, this will not suit the client's models as the usage in the clinical field will warrant a lot of useful time and resources which will expand the expense separately. We can close by expressing that the HGA model outperforms the traditional and other AI models, and as a base, conservative expense differentiation from other contrasted models.

Moreover, how the tested model will assume an urgent part in the clinical field where assets are utilized continuously is an eminent concern. However, in Table 7, we depict the characteristics and the QoS of each model. The table shows the HGA and other models with a significant QoS. This shows that cloud providers are expected to accomplish maxima income while thinking about QoS and solicitations from the clients. The Healthcare 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 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 answer approaching solicitations without the apprehension about a task being dissolved or terminated.

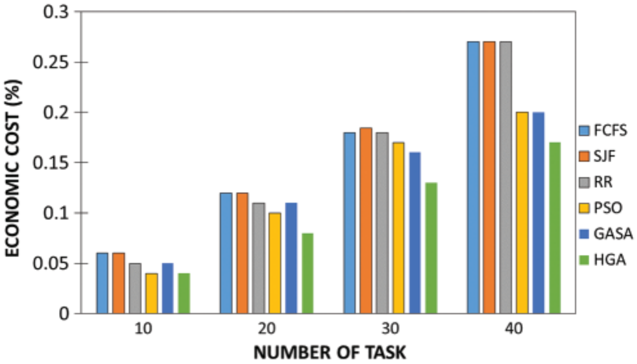

Figure 11: Scheduling models *vs.* economic cost

Table 7: The resulting QoS, Pros, and Cons

Scheduling model	Cons	Pros	QoS
SJF	Complexity in comprehending.	The task with minimum execution time is scheduled first.	It has a medium waiting time and the highest execution time.
FCFS	No more scheduling criteria.	Implementation is straightforward.	Little finish time, little waiting time, and little execution time.
RR	It requires pre-emption.	Efficient balanced tasks with less complexity.	Least waiting time, Small execution time, and smallest finish time.
PSO	Low-quality solution.	The tuning parameter is less.	Certification is on a medium scale.
GASA	Complexity in coding.	Its efficiency is on simulated annealing.	Better execution and waiting time compared to the traditional model.
HGA	Complexity in coding and understanding.	It is based on various decision criteria, crossover, mutation, and fitness function.	The best throughput. With the smallest execution time.

7 Conclusion

The proposed work put 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 the most invigorating point for researchers, industries, and public zones. The goal of this work is to style a model that can solve task scheduling issues while, sharing resources to reach a productive QoS. Thus, this theory targets developing a fast, sharp, and particular structure for task scheduling for IoMT-cloud. This evaluation relies upon utilizing present-day developments to further develop research on IoMT-cloud. This work presents an overall report between the traditional optimization techniques and other AI techniques in IoMT-cloud, like the SJF, FCFS, RR, PSO, GASA, and the proposed model being the HGA. Several parameters were also used for validation and comparison, like the TWT, TFT, TFT, resource utilization, throughput, and cost. This work re-authorized the proposed assessment with various scheduling techniques to display the ampleness of the HGA. This work will offer a potential manual for clients and experts during IoMT-cloud execution. From the diagrams and calculations, it was exhibited that the HGA outperformed 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 processes in cloud computations. The proposed HGA had an execution pace of 32.47 ms and a throughput of 1.23 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 the traditional and other AI optimization models 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. IoMT requires high execution speed and is time-dependent. However, dependent upon the actions while setting up this assessment, the future examination is to be considered like completing the computation for other progression factors like speedup and streamlining time. In future work, we can lessen the cost and increase throughput with the computations to get more smoothed results. Finally, we will update the work using several other characteristics and bring the outcomes as they appear. We are experimenting with more AI models like the bee algorithm and Ant algorithm. This is provided to work on the sufficiency of task scheduling in the IoMT-cloud. 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.

Supplementary Materials: The authors declare no supplementary material was used in this study.

Funding Statement: The authors declare that no funding was received from any financial organizations regarding the material reported in this manuscript.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Al-Turjman, F., Hussain, A. A., Alturjman, S., Altrjman, C. (2022). Vehicle price classification and prediction using machine learning in the IoT smart manufacturing era. *Sustainability, 14(15),* 9147. DOI 10.3390/su14159147.
2. Calheiros, R. N., Ranjan, R., Beloglazov, A. (2011). CloudSim: A toolkit for modelling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience, 41(1),* 23–50.
3. Zhu, Z., Du, Z. (2013). Improved GA-based task scheduling algorithm in cloud computing. *Computer Engineering and Applications, 49(5),* 77–80.
4. Zhou, L., Wang, C. (2015). Cloud computing resource scheduling in mobile internet based on particle swarm optimization algorithm. *Computer Science, 42(6),* 279–292.
5. Xu, J., Zhu, J., Lu, K. (2013). Task scheduling algorithm based on dual fitness genetic annealing algorithm in cloud computing environment. *Journal of University of Electronic Science and Technology of China, 42(6),* 900–904.

6. Wang, F., Li, M., Daun, W. (2013). Cloud computing task scheduling based on dynamically adaptive ant colony algorithm. *Journal of Computer Applications, 33(11),* 3160–3162. DOI 10.3724/SP.J.1087.2013.03160.
7. Al-Turjman, F., Deebak, D. (2020). Privacy-aware energy-efficient framework using internet of medical things for COVID-19. *IEEE Internet of Things Magazine, 3,* 64–68. DOI 10.1109/IOTM.0001.2000123.
8. Subramani, P., Al-Turjman, F., Kumar, R., Kannan, A., Loganthan, A. (2021). Improving medical communication process using recurrent networks and wearable antenna s11 variation with harmonic suppressions. *Personal and Ubiquitous Computing, 3,* 1–13. DOI 10.1007/s00779-021-01526-3.
9. Rajalingam, B. (2020). Intelligent multimodal medical image fusion with deep guided filtering multimedia systems. *Multimedia Systems, 28,* 1449–1463. DOI 10.1007/s00530-020-00706-0.
10. Reynolds, R. G., Michalewicz, Z., Cavaretta, M. (1995). Using cultural algorithms for constraint handling in GENOCOP. *Procceding of the 4th Annual Conference on Evolutionary Programming*, pp. 298–305. Cambrige, MIT Press.
11. Hussain, A. A., Bouachir, O., Al-Turjman, F., Aloqaily, M. (2020). AI techniques for COVID-19. *IEEE Access, 8,* 128776–128795. DOI 10.1109/ACCESS.2020.3007939.
12. Hussain, A. A., Al-Turjman, F. (2020). Resource allocation in volunteered cloud computing and battling COVID-19. In: *AI-Powered IoT for COVID-19*, pp. 39–76. Boca Raton: CRC Press. DOI 10.1201/9781003098881-2.
13. Al-maamari, A., Omara, F. (2015). Task scheduling using PSO algorithm in cloud computing environments. *International Journal of Grid Distribution Computing, 8(5),* 245–256.
14. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. et al. (2010). A view of cloud computing. *Communication ACM, 53(4),* 50–58. DOI 10.1145/1721654.1721672.
15. Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y. C., Talbi, E. G. et al. (2011). A parallel bi-objective hybrid meta heuristic for energy-aware scheduling for cloud computing systems. *Journal of Parallel Distributed Computing, 71(11),* 1497–1508. DOI 10.1016/j.jpdc.2011.04.007.
16. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT), a vision, architectural elements, future directions. *Future Generation Computing System, 29(7),* 1645–1660. DOI 10.1016/j.future.2013.01.010.
17. Scarlett, R. (2019). The IoT trends that no one has spoken about-read this now. https://towardsdatascience.com/top-14-iot-trends-to-expect-in-2020-fa81a56e8653.
18. Li, Q., Guo, Y. (2010). Optimization of resource scheduling in cloud computing. *2010 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, vol. 8, pp. 315–320.
19. Singh, P., Dutta, M., Aggarwal, N. (2020). A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowledge and Information Systems, 62,* 1–51. DOI 10.1007/s10115-018-1254-2.
20. Baker, T., García-Campos, J. M., Reina, D. G., Toral, S., Taw, H. et al. (2018). GreeAODV: An energy efficient routing protocol for vehicular ad hoc networks. *Proceeding International Conference of Intelligent Computing*, pp. 670–681. Cham, Switzerland, Springer.
21. Ridhawi, I. A., Aloqaily, M., Kotb, Y., Jararweh, Y., Baker, T. (2021). A profitable and energy-efficient cooperative fog solution for IoT services. *IEEE Transactions on Industrial Informatics, 16(5),* 3578–3586.
22. Oueida, S., Kotb, Y., Aloqaily, M., Jararweh, Y., Baker, T. (2018). An edge computing based smart healthcare framework for resource management. *Sensors, 18(12),* 4307. DOI 10.3390/s18124307.
23. Al-Khafajiy, M., Baker, T., Al-Libawy, H., Maamar, Z., Aloqaily, M. et al. (2019). Improving fog computing performance via fog-2-fog collaboration. *Future Generation Computing System, 100,* 266–280. DOI 10.1016/j.future.2019.05.015.
24. Kotb, Y., Al Ridhawi, I., Aloqaily, M., Baker, T., Jararweh, Y. et al. (2019). Cloud-based multiagent cooperation for IoT devices using workflow nets. *Journal of Grid Computing, 4,* 1–26. DOI 10.1007/s10723-019-09485-z.

25. Sangwan, P., Sharma, M., Kumar, A. (2017). Improved round robin scheduling in cloud computing. *Advances in Computational Sciences and Technology, 10(4),* 639–644.

26. Li, J., Ma, T., Tang, M., Shen, W., Jin, Y. (2017). Improved FIFO scheduling algorithm based on fuzzy clustering in cloud computing. *Information, 8(1),* 25. DOI 10.3390/info8010025.

27. Elmougy, S., Sarhan, S., Joundy, M. (2017). A novel hybrid of shortest job first and round robin with dynamic variable quantum time task scheduling technique. *Journal of Cloud Computing, 6(1),* 1–12.

28. Thakur, P., Mahajan, M. (2017). Different scheduling algorithm in cloud computing: A survey. *International Journal of Modern Computer Science, 5(1),* 68–72.

29. Inui, S. (2020). Chest CT findings in cases from the cruise ship 'diamond princess' with coronavirus disease 2019 (COVID-19). *Radiology Cardiothoracic Image,* 2*(2),* 110–200.

30. Arkhipov, D. I., Wu, D., Wu, T., Regan, A. C. (2020). A parallel genetic algorithm framework for transportation planning and logistics management. *Access IEEE, 8,* 106506–106515. DOI 10.1109/Access.6287639.

31. Gan, Y., Yin, C., Fan, Q., Li, A. (2020). Improved T-matrix method for simultaneous reconstruction of dielectric and perfectly conducting scatterers. *Access IEEE, 8,* 143622–143631. DOI 10.1109/Access.6287639.

32. Tsai, J. T., Fang, J. C., Chou, J. H. (2013). Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *Computer Operational Research, 40(12),* 3045–3055. DOI 10.1016/j.cor.2013.06.012.

33. Maguluri, S. T., Srikant, R. (2014). Scheduling jobs with unknown duration in clouds. *IEEE/ACM Transactions on Networking, 22(6),* 1938–1951. DOI 10.1109/TNET.2013.2288973.

34. Cheng, C., Li, J., Wang, Y. (2015). An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. *Tsinghua Science and Technology, 20(1),* 28–39. DOI 10.1109/TST.2015.7040511.

35. Lin, W., Liang, C., Wang, J. Z., Buyya, R. (2014). Bandwidth-aware divisible task scheduling for cloud computing. *Software: Practice and Experience, 44(2),* 163–174. DOI 10.1002/spe.2163.

36. Ergu, D., Kou, G., Peng, Y., Shi, Y., Shi, Y. (2013). The analytic hierarchy process: Task scheduling and resource allocation in cloud computing environment. *The Journal of Supercomputing, 64(3),* 835–848. DOI 10.1007/s11227-011-0625-1.

37. Zhu, X., Yang, L. T., Chen, H., Wang, J., Yin, S. et al. (2014). Real-time tasks-oriented energy-aware scheduling in virtualized clouds. *IEEE Transactions on Cloud Computing, 2(2),* 168–180. DOI 10.1109/TCC.2014.2310452.

38. Liu, X., Zha, Y., Yin, Q., Peng, Y., Qin, L. (2015). Scheduling parallel jobs with tentative runs and consolidation in the cloud. *Journal of System Software, 104,* 141–151. DOI 10.1016/j.jss.2015.03.007.

39. Shamsollah, G., Othman, M. (2012). Priority based job scheduling algorithm in cloud computing. *Procedia Engineering, 50,* 778–785.

40. Rodriguez, M. A., Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workows on clouds. *IEEE Transactions on Cloud Computing, 2(2),* 222–235. DOI 10.1109/TCC.2014.2314655.

41. Polverini, M., Cianfrani, A., Ren, S., Vasilakos, A. V. (2014). Thermal aware scheduling of batch jobs in geographically distributed data centers. *IEEE Transactions on Cloud Computing, 2(1),* 71–84. DOI 10.1109/TCC.2013.2295823.

42. Keshk, A. E., El-Sisi, A. B., Tawfeek, M. A. (2014). Cloud task scheduling for load balancing based on intelligent strategy. *International Journal of Intelligent System Application, 6(5),* 25–36.

43. Ghanbari, S., Othman, M., Leong, W. J., Bakar, M. R. A. (2014). Multi-criteria-based algorithm for scheduling divisible load. *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, vol. 285, pp. 547–554. Singapore.

44. Goudarzi, H., Ghasemazar, M., Pedram, M. (2012). Sla-based optimization of power and migration cost in cloud computing. *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012)*, pp. 172–179. IEEE Computer Society, Ottawa, Canada.

45. Al-Turjman, F., Alturjman, S. (2020). 5G/IoT-enabled UAVs for multimedia delivery in industry-oriented applications. *Multimedia Tools Application, 79,* 8627–8648. DOI 10.1007/s11042-018-6288-7.

46. Alabady, S. A., Al-Turjman, F., Din, S. (2020). A novel security model for cooperative virtual networks in the IoT era. *International Journal of Parallel Programming, 48,* 280–295. DOI 10.1007/s10766-018-0580-z.

47. Al-Turjman, F. (2020). Intelligence and security in big 5G-oriented IoNT: An overview. *Future Generation Computer Systems, 102,* 357–368. DOI 10.1016/j.future.2019.08.009.

48. Ghanbari, S., Othman, M., Bakar, M. R. A., Leong, W. J. (2015). Priority-based divisible load scheduling using analytical hierarchy process. *Applied Mathematical Information Science, 9(5),* 25–41.

49. Radojevic, B., Zagar, M. (2011). Analysis of issues with load balancing algorithms in hosted (cloud) environments. *2011 Proceedings of the 34th International Convention*, pp. 416–420. Opatija, Croatia.

50. Ghanbari, S., Othman, M., Bakar, M. R. A., Leong, W. J. (2016). Multi-objective method for divisible load scheduling in multi-level tree network. *Future Generation Computing System, 54,* 132–143. DOI 10.1016/j.future.2015.03.015.

51. Goswami, S., Das, A. (2017). Optimization of workload scheduling in computational grid. *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, pp. 417–424. Odisa, India.

52. Kaur, K., Kaur, A. (2016). A hybrid approach of load balancing through VMs using ACO, MinMax and genetic algorithm. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 615–620. Dehradun.

53. Pilavare, M. S., Desai, A. (2015). A novel approach towards improving performance of load balancing using genetic algorithm in cloud computing. *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–4. Coimbatore.

54. Patel, R. R., Patel, S. J., Patel, D. S., Desai, T. T. (2016). Improved GA using population reduction for load balancing in cloud computing. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2372–2374. Jaipur.

55. Farrag, A. A. S., Mahmoud, S. A., El-Horbaty, E. S. M. (2015). Intelligent cloud algorithms for load balancing problems: A survey. *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 210–216. Cairo.

56. Gharehchopogh, F. S. (2022). An improved tunicate swarm algorithm with best-random mutation strategy for global optimization problems. *Journal of Bionic Engineering, 19,* 1177–1202. DOI 10.1007/s42235-022-00185-1.

57. Abedi, M., Gharehchopogh, F. S. (2020). An improved opposition based learning firefly algorithm with dragonfly algorithm for solving continuous optimization problems. *Intelligent Data Analysis, 24(2),* 309–338. DOI 10.3233/IDA-194485.

58. Mohammadzadeh, H., Gharehchopogh, F. S. (2021). A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study email spam detection. *Computational Intelligence, 37(1),* 176–209. DOI 10.1111/coin.12397.

59. Mohmmadzadeh, H., Gharehchopogh, F. S. (2021). An efficient binary chaotic symbiotic organisms search algorithm approaches for feature selection problems. *The Journal of Supercomputing, 77(8),* 9102–9144. DOI 10.1007/s11227-021-03626-6.

60. Ghafori, S., Gharehchopogh, F. S. (2021). Advances in spotted hyena optimizer: A comprehensive survey. *Archives of Computational Methods in Engineering, 3,* 1569–1590.

61. Makasarwala, H. A., Hazari, P. (2016). Using genetic algorithm for load balancing in cloud computing. *2016 8th International Conference on Electronics Computers and Artificial Intelligence (ECAI)*, pp. 1–6. Ploiesti.

62. Kavitha, K. V., Suthan, V. V. (2016). Dynamic load balancing in cloud-based multimedia system with genetic algorithm. *2016 International Conference on Inventive Computation Technologies (ICICT)*, pp. 1–4. Coimbatore.

63. Dam, S., Mandal, G., Dasgupta, K., Dutta, P. (2015). Genetic algorithm and gravitational emulation-based hybrid load balancing strategy in cloud computing. *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pp. 1–7. Hooghly.

64. Liang, H., Ge, Y. F. (2015). GACA-VMP: Virtual machine placement scheduling in cloud computing based on genetic ant colony algorithm approach. *2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing and 2015 IEEE 12th International Conference on Autonomic and Trusted Computing and 2015 IEEE 15th International Conference on Scalable Computing and Communications and its Associated Workshops (UIC-ATC-ScalCom)*, pp. 1008–1015. Beijing.

65. Hussain, A. A., Al-Turjman, F. (2021). Artificial intelligence and blockchain: A review. *Transactions on Emerging Telecommunications Technologies, 32,* e4268. DOI 10.1002/ett.4268.

66. Manzoor, A., Shah, M. A., Khattak, H. A., Din, I. U., Khan, M. K. (2019). Multi-tier authentication schemes for fog computing: Architecture, security perspective, challenges. *International Journal of Communication Systems, 35,* e4033. DOI 10.1002/dac.4033.

67. Hussain, A. A., Al-Turjman, F., Sah, M. (2021). Semantic web and business intelligence in big-data and cloud computing era. *Computer Science, 4,* 1418–1432. DOI 10.1007/978-3-030-66840-2_107.

68. Hussain, A. A., Al-Turjman, F., Gemikonakli, E., Ever, Y. K. (2021). Design of a navigation system for the blind/visually impaired. In: Ever, E., Al-Turjman, F. (eds.), *Forthcoming networks and sustainability in the IoT era*, vol. 353. Springer, Cham. DOI 10.1007/978-3-030-69431-9_3.

69. Hussain, A. A., Dimililer, K. (2021). Student grade prediction using machine learning in IoT Era. *Forthcoming Networks and Sustainability in the IoT Era*, *353,* 65–81. DOI 10.1007/978-3-030-69431-9_6.

70. Almogren, A., Mohiuddin, I., Din, I. U., Almajed, H., Guizani, N. (2021). FTM-IoMT: Fuzzy-based trust management for preventing sybil attacks in internet of medical things. *IEEE Internet of Things Journal, 8(6),* 4485–4497. DOI 10.1109/JIOT.2020.3027440.

71. Gharehchopogh, F. S., Farnad, B., Alizadeh, A. (2021). A modified farmland fertility algorithm for solving constrained engineering problems. *Concurrency and Computation: Practice and Experience, 33(17),* e6310. DOI 10.1002/cpe.6310.

72. Zaman, H. R. R., Gharehchopogh, F. S. (2021). An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems. *Engineering with Computers, 2(2),* 1–35. DOI 10.1007/s00366-021-01431-6.

73. Gharehchopogh, F. S., Abdollahzadeh, B. (2021). An efficient harris hawk optimization algorithm for solving the travelling salesman problem. *Cluster Computing, 25,* 1981–2025.

74. Mohammadzadeh, H., Gharehchopogh, F. S. (2021). Feature selection with binary symbiotic organisms search algorithm for email spam detection. *International Journal of Information Technology & Decision Making, 20(1),* 469–515. DOI 10.1142/S0219622020500546.

75. Mohammadzadeh, H., Gharehchopogh, F. S. (2021). A multi-agent system based for solving high-dimensional optimization problems: A case study on email spam detection. *International Journal of Communication Systems, 34(3),* e4670. DOI 10.1002/dac.4670.