Solving a task scheduling problem in systems based on fog calculations by presenting a prohibiting search base metaheuristic algorithm

1 ALI MOEINYAVARI,2 SEPIDEH SARAYLOO, 3 MAHTAB ILTARABIAN

1, Master degree in Software Engineering, Islamic Azad University, Science and Research Branch, Iran

2, PhD candidate in Software Engineering, Islamic Azad University, North Tehran Branch, Iran

3, PhD candidate in Software Engineering, Islamic Azad University, North Tehran Branch, Iran

alimoien.y@gmail.com, sepideh.sarayloo@gmail.com, m.iltarabian@iau-tnb.ac.ir

Abstract

Nowadays, due to the expansion of communications, the volume of data has increased and subsequently there is more need to enhance the speed of accountability. Large volume of data processing will require processing infrastructure as well as higher storage, which is quite expensive. In this study, we aim to reduce the response time and reduce the cost of cloud computing and fog. One of the most significant issues in this matter is the issue of resource allocation. So, we can achieve load balance in addition to the increase level of productivity. The prohibited search has turned it into a new metaheuristic method among the various metaheuristic methods, due to its great expansion in various optimization problems as well as the characteristics of memory and high speed. So, higher performance can be compared to similar algorithms. In the current study, a new metaheuristic method is proposed which is optimized by the nearest neighbor algorithm.

Keywords: Fog Calculations, Task Scheduling, Forbidden Search, Fog-Cloud Platform, Smart Home Energy Management

Tob Regul Sci. ™ 2023;9(1): 846-872
DOI: doi.org/10.18001/TRS.9.1.60

Introduction

In today's modern life, cost factor has become one of the most important goals of organizations and companies because humans tend to consume or produce at low cost and high quality. Therefore, effective methods and strategies are needed in various fields to manage and reduce costs. On the other hand, the importance of time factor and the speed of operations is increased due to the increasing advancement of technology, the expansion of communications as well as the expansion of smart cities. Consequently, it increases the data production in the world. In order to achieve these two goals and reduce them simultaneously, systems with infrastructures that can support these two factors at the same time should be created. Also, in order to create these infrastructures, platforms that form communications and interactions and define the rules and rules governing these interactions are needed. Therefore, designing these platforms will bring us closer to achieve these two goals.

In recent years, common and popular technology for creating intelligent systems and improving communications and interactions has been the cloud that created a great revolution in industry and technology. In fact, cloud computation created a platform with high computational and storage capacities for these communications and interactions. But with increasing data production, the processing time of operations on them and delay in the network also increased. Due to this reason, fog computing technology was used in combination with cloud computing, which by approaching the user and smart internet devices of objects reduced the operation time and delay of processes. Therefore, the current technologies for creating such systems, cloud computing and fog diene in combination with smart internet devices are the basic objects. As a result, researchers are looking to design and implement infrastructures and platforms using these technologies that can follow the target factors simultaneously. In addition to this, cloud computing, fog speaking and combining them with smart devices describe the purpose and summary of this article. The cloud-based platforms are expanding almost widely as powerful and accessible digital technologies are growing fastly. Moreover, for the sake of better distribution of resources, integration of cloud computing with high performance smart grid, secure transmission and power distribution, continuous management system, scalable, economical and flexible features are good. Cloud computing dynamically assigns computational, communication and efficient virtual storage resources to objects' internet devices. Cloud computing provides services based on the user's needs, regardless of their geographical location, which is classified into three categories: software, hardware and platform. [1] The main idea behind this technology is that hardware and software resources can be provided by virtualizing and sharing resources from the Internet. As a result, hardware and software purchase fees and maintenance costs for users are eliminated. This is one of the reasons for the importance and popularity of this technology in recent years. Although cloud computing has been a significant infrastructure for this purpose, and large internet data of objects produced by smart home systems and smart sensors require real-time processing and analysis. The cloud-based architectures process large volumes of data with high latency and high complexity.

Additionally, the cloud core is often physically or reasonably far from final users, pointing out that communications and data transmission must pass through many hubs, which generate a lot of delay and occupy network bandwidth. [2] Therefore, another possible technology is needed in which computational servers are located near the final physical user at the edge of the network in order to collect data, to calculate the sent tasks, perform the requested services, and major analysis. Fog computational nodes are resource efficient because they are equipped with virtual machine technologies that can continuously process internet data streams of new objects and then transfer this processed data to the cloud for further processing [3]. Cloud computational method offers several advantages, such as infrastructure as a service to access the unlimited storage space.

Platform as a Service has potential to run effective applications in terms of resources, software as services: Software access facilities and utility services: storing large amounts of data for remote access. Fog computing system plays a vital role in objects' internet system to support big data processing for near-real-time responses. Fog computational method also basically processes and stores data at the edge of the cloud system [4]. This integrated architecture allows us to resolve latency problems related to the cloud system's basic transmission communication network, which has a special impact on time-sensitive applications. Integration of big data analysis of internet objects with fog-cloud architecture requires the platform as an interface between consumers and manufacturers.[5] Several architectures remain in the concept of fog-cloud systems- allocation of resources and internet objects 1: fog and optimization, calculations without servers, energy consumption, data management and system locality, coordination in fog nodes, dynamics, urban calculations, internet of industrial objects, etc. [6] Inside the fog node, a large number of virtual machines are created on servers and hosts. Therefore, when a service is requested by the end user, it is divided into a set of tasks to run on virtual machines. Allocating thousands of fog nodes to 10,000 tasks requires a favorable scheduling algorithm and this will define the job scheduling problem [7] In this article, according to figure (1), the goal is to design and implement a platform using cloud and fog computing technologies and in combination with smart devices for smart city that can reduce costs and time while performing high processing speeds and we can use in various fields and systems of smart city. To implement this platform, a case study of smart home with real data in Tehran has been simulated. This platform consists of four elements:

1- Smart home appliances

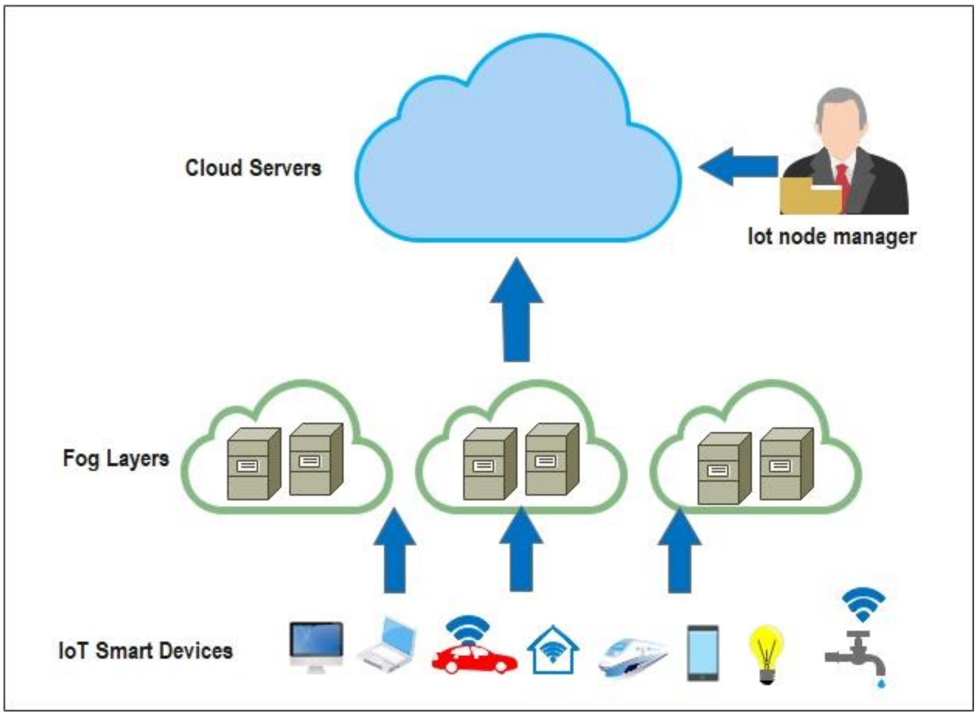
2- Management system of IoT services

3- Nodes of Fog

4- Cloud system

To achieve the desired results, the number of fog layers between the cloud system and smart devices according to the two factors of the target cost and response time, is discussed and the most appropriate number is determined, so that both the cost and the time of the imagination as much low as possible. Therefore, the minimum cost and response time are obtained. Another issue that has been discussed in this article. Nowadays as one of the challenging problems in such systems is the scheduling problem of tasks between fog nodes and among thousands of virtual machines that are simulated and implemented in the proposed platform and in the case study.

Figure 1 Elements of the proposed platform to create the desired infrastructure



In fact, the most important advantage of using task scheduling algorithm is assigning a large number of tasks to the lowest number of fog nodes that are using less memory and processor execution time. In addition to this, the allocation of fog resources to the desired tasks is controlled in which the load of the system is balanced, energy consumption and response time are minimized and the processing speed of the tasks increases. Since there is no precise algorithm for solving task scheduling problems, it is considered as a difficult optimization problem. Therefore, the objectives of the problem are summarized:

1- Providing a fog-cloud-based platform by evaluating the number of different Fog layers to obtain the appropriate number of layers according to the cost factors and response time.

2- Scheduling tasks among virtual machines in fog nodes using an improved metaheuristic algorithm based on prohibited search

In the continuation of this article, first, thematic literature and related concepts were studied which includes the platform based on cloud and fog calculations and its constituent elements, task scheduling, scheduling algorithms presented in cloud and fog calculations, and algorithms. The proposed proposal is expressed, then the timing of independent tasks is implemented immediately based on the proposed algorithm in the proposed computational platform. Finally, a comparison is made in the same conditions between the base algorithm and the proposed algorithm.

Maximum Research:

Several studies deal with issues similar to the work of this study, a limited number of which are mentioned in this article, such as [7,8,12,13,14 united platforms] that perform complex operations on data received from internet smart devices, this study addresses the challenges of resource management for classification, processing and storage of internet data in online and offshore modes of components. The platforms in their work include smart core dials, internet management of objects, and integration of services and nodes of fog computing and cloud system. [7] They have investigated the problem of scheduling tasks in the fog computing paradigm, which aims to provide high-efficiency and cost-effective services. They proposed a new bio-optimization algorithm called Bee Life Algorithm 1 to allocate a set of job tasks among the fog nodes. In addition to this, they also compared their study to optimization of particle swarms and genetic algorithms. [13] They proposed a four-layer big data analysis platform integrated with Fog's calculations to analyze the energy consumption data of homehold appliances. Their platform includes the acquisition of internet data, internet management of objects and the integration of services, fog computing nodes and the cloud system. The platform supports huge data with complex operations and ever-available applications. Furthermore, they have provided a data analysis engine on their platform for short-term analysis on the cloud system. [14] They focused on the issue of allocating virtual machines in a cloud-fog model. The authors proposed two methods based on semi-Markov decision-making to reduce the cost of providing services by remote cloud and to balance the limited computational capacity of local fog nodes. [12] They also proposed a rule-based algorithm to minimize response time, in which it assigns priorities to tasks and resources from fog and cloud. Also, their proposed method collects available free resources with a new concept of free resource space to in order to prevent the occurrence of bombast. Mohammad pur and Parvin presented a new dynamic optimization algorithm based on the chaotic artificial bee colony with memory and evaluated it on the benchmark problem of moving peaks [16]. In the same subject, Hosseini and Hassani have developed a branch and bound algorithm for scheduling the processing of parts in the first station and assembling products at the second station in the workshop flow generation system, so that the completion time is minimized [1] In another context, Sharifzadeh and Amjadi have used the particle sectorial optimization algorithm for optimal distribution of reactive power and compared them with genetic algorithm [18]

Research methodology:

In this study, according to the proposed platform for creating the infrastructure As per the shape, the fog layer is composed of fog nodes located in different parts of a home (such as bedroom, kitchen, living room, dining room and bathroom) and are close to smart electronics. Therefore, homehold appliances in each part of the home are connected to their local fog node. Also, depending on the geographical location in the upper layer each region of a city has local nodes or nodes that are located in the second layer in this architecture and the fog node of the home is related to the fog node of its area. Finally, this layer is connected to the energy organization cloud. According to the objective factors of cost, response time, number of fog layers are determined between cloud system and internet objects and different assumptions are compared with each other. In this proposed model, the node of internet manager of objects located in the cloud layer performs the proposed scheduling algorithm for allocating all the sent tasks. To find the optimal timing, a metacognitive health algorithm, smart homes, etc. is needed in which the response time for processing should be predicted close to the real time. On the other hand, there are many limitations in internet data processing of objects and calling cloud services. The proximity of resources helps to minimize system latency, which is related to the preparation of cloud services, which are implemented by the node of the internet administrator of objects.

Table 1 of the most important recent researches, similar to the work of the current article

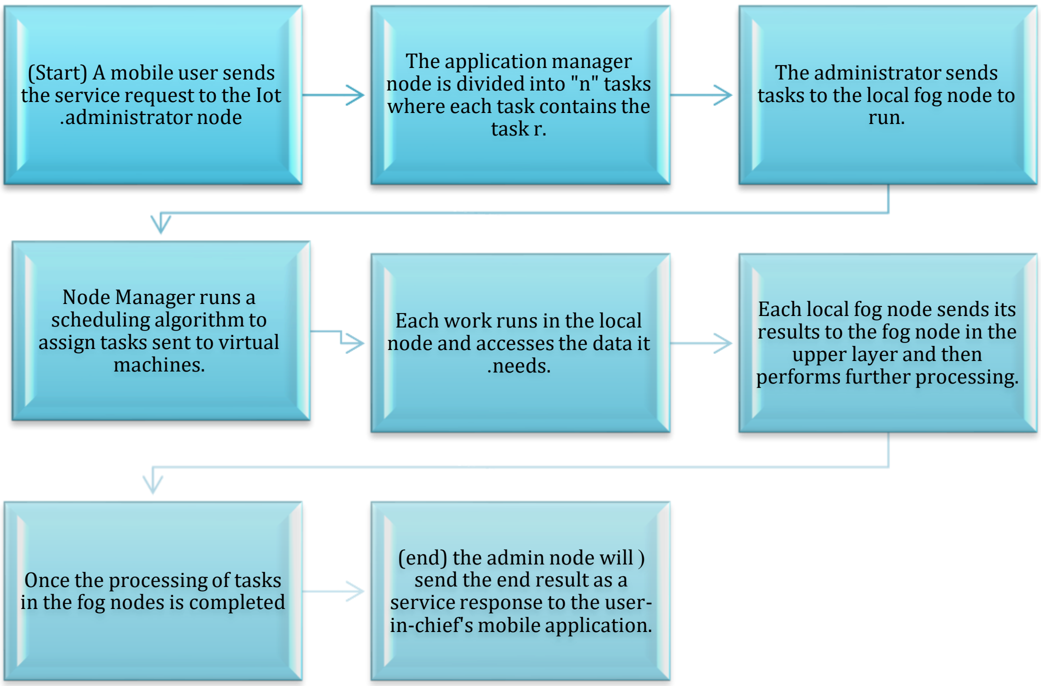
The appropriate number of fog layers is not specified	Scalable analysis virtualization Distribution of resources	Scheduling in a four-layer platform based on Fog-cloud	[8] A. Yassine, S. Singh, M.S. Hossain and G. Muhammad
Execution time, response time and dedicated memory are not taken into account.	High performance Affordable services Comparison with PSO and GA algorithms	Task scheduling problem in fog computing, proposal of BLA algorithm	[7] S. Bitam, S. Zeadally and A. Mellou
Delay, cost and execution time are not considered.	Computationally intensive analysis access Short term analysis	Four-layer platform based on Fog-Cloud-Iot	[13] S. Singh and A. Yassine
Execution time, response time and allocated memory are not taken into account.	Reducing the cost of providing services Capacity balance of FOG nodes	Allocation of virtual machines in a Fog-Cloud model	[14] [14] Q. Li, L. Zhao, J. Gao, H. Liang, L. Zhao and X. Tang

Cost and allocated memory are not taken into account.	Minimizing response time, energy consumption and latency Avoid deadlocks	Task scheduling in Fog-Cloud model with rule-based algorithm	[12] S.K. Sood and K.D. Singh
It has not been compared with any other algorithm.	Minimize time	The combination of tabu-search and variable neighborhood descent algorithms	[15]] Z. Zeng, X. Yu, K. He and Z. Fu

Timing and Tasks:

Smart devices and home appliances produces a large amount of data that requires analytical methods that are cost effective in terms of resources and cost. Each fog node is composed of data centers and each data center has at least one or more physical hosts. Virtual machines are also created separately on fizzy hosts that share resources, and these machines must perform the tasks requested by a user on their host and provide the result. Therefore, it is possible to receive multitasking simultaneously, so each virtual device must have a scheduling mechanism. Therefore, scheduling tasks for data mining of internet objects are required for various smart city applications, such as automated demand response, health and wellness programs, smart homes, etc. in which response time for processing should be predicted close to real time. On the other hand, there are many limitations in the processing of objects' internet data and the calling of cloud services. The proximity of resources helps to minimize system latency, which is related to the preparation of cloud services. Therefore, the coordination of the tasks provided among the nodes indicating the services requested by mobile users is located at the edge of the network. Moreover, assigning these tasks to cloud system resources appropriately requires optimization of task scheduling mechanisms. The task scheduling problem is used to minimize the processor execution time and memory used by the tasks provided. In the next section, the proposed system model for the cloud-fog platform is presented and the task scheduling problem will be explained. Therefore, the fog processing technology also reduces the processing time and memory allocated to the requested services. The operation of the proposed model by explaining the steps to run the required service is as follows in Figure 2.

Figure 2- Stages of scheduling by IoT manager node



The mathematical expression of the problem

In this section, it explains the problem of scheduling tasks in the fog computational environment. When a user submits a service request to the system, the request can be defined as a set of occupations. Then, the work is set to process and run to the n-system, which is defined as follows:

$$\text{Jobs} = \{J_1 . J_2 \dots \dots . J_i \dots \dots J_n \qquad (1)$$

Each work i of the work set (1<i<n) can be divided into a set of tasks (ie 1<k<r) and this is sent to the node j.

Jobi Tasks={JTaskji1.JTaskji2…………….JTaskjik………….JTaskjir} (2)

For example i work tasks:

$$\text{Job}_i \text{ Tasks} = \{\text{JTask}_{i1}{}^{2} . \text{JTask}_{i2}{}^{2} . \text{JTask}_{i3}{}^{2}\,\} \qquad (3)$$

The first task (JTask2i1) and the second task (JTask2i2) and the third task (JTask2i3) of the work I are sent to the fog 2 node. So each local node receives its own work with a set of tasks. FNj includes the tasks of the job in question as follows.

$$\mathrm{FN_i\ Tasks} = \{\mathrm{JTask_{ax}}^{j} \,.\, \mathrm{JTask_{by}}^{j} \ldots\ldots \mathrm{JTask_{ik}}^{j} \ldots\ldots \mathrm{JTask}\} \qquad (4)$$

In addition, each FNj node has a large number of virtual machines, VMm (1 ≤ the number of VMs ≤ m) created on its hosts and tasks are sent to it to run. The problem is the scheduling of the tasks of these virtual machines in the fog node.

$$FNjVms = \{VM1 \ldots\ldots\ldots VMm\} \qquad (5)$$

After scheduling, tasks sent to the virtual machine in VMm are assigned in the FNj fog node.

$$FNjVMm = \{JTask13\ jm.\ JTask57\ jm.\ JTask84\ jm\} \qquad (6)$$

So VMm performs the third task of work 1, the seventh task of work 5 and the fourth task of work 8 in the FNj node. In addition, the total cpu execution time to assign tasks (r tasks) to FNj is as follows:

$$CPU\_Execution\_Time(FNjTasks) = (JTaskik\ jm\ .\ StartTime + JTaskik\ jm\ .\ ExeTime)\ sum$$

$$1 \leq k \leq r$$

$$\boldsymbol{i \in jobs\ of\ selected\ tasks} \qquad (7)$$

JTaskjmik.Start Time shows when the k task starts from an i job, in VMm and FNj. JTaskikj.ExeTime CPU is the time to run K task in FNj. In addition, the memory size required to assign the k task in FNj is calculated as follows.

$$Memory(FNj\ Tasks) = (JTaskik\ jm\ .\ AllocatedMemory) \quad max \quad 1 \leq k \leq r$$

$$\boldsymbol{i \in jobs\ of\ selected\ tasks} \qquad (8)$$

Finally, the question of scheduling tasks in the node of fog can be described as follows:

$$\boldsymbol{FNTasks} = \{\boldsymbol{FN1Tasks}.\boldsymbol{FN2Tasks}\ldots\ldots\ldots\boldsymbol{FNmTasks}\} \qquad (9)$$

$$\boldsymbol{FNjTasks} = \{\boldsymbol{JTaskaxj1}\ .\boldsymbol{JTaskbyj2}\ldots\ldots\ldots\ldots\ldots\ldots\boldsymbol{JTasknrjm}\} \qquad (10)$$

3-3. The cost of defining a cost function is necessary to evaluate the quality of the proposed solution. This cost function is a minimization function for calculating the desired value of the following objectives: processor execution time and assigned memory size.

$$\mathrm{Cost\_function(FNTasks)} = \mathrm{Min}\left[\sum_{m}^{j=1} \left(\mathrm{Cost_{function}}\ \mathrm{JTask}_{ik}^{jm}.\mathrm{FN_j}\right)\right] \quad (11)$$

$$\mathrm{Cost\_function(FNTasks)(JTask_{ik}^{jm}.FN_j)} = \mathrm{w1.CPU\_Execution\_Time(FN_jTasks)} + \mathrm{w2.Memory\big(FN_jTasks\big)} \qquad (12)$$

w1 and w2 are important coefficients that are decision makers and also define on the basis of their preferences of these two objective factors (i.e. the time of implementation of the assigned processor and memory[6]

combinations.

Metaheuristic algorithms have not been used by researchers in the form of single-axis methods in the past decades, but the combined methods of other classes of optimization algorithms have been considered as an important issue. [19] These approaches are generally called metaheuristic combinations 1. Combining different algorithms with different concepts creates high-efficiency systems that simultaneously exploit the benefits of individual algorithms. The efficiency of these hybrid methods is due to collaboration between algorithms. Therefore, a suitable combination of metaheuristic algorithms is needed to solve optimization problems in a cost-effective and efficient manner. In fact, an exclusive algorithm regulated from the appropriate combination of a metaheuristic algorithm and other algorithmic techniques is needed to solve a specific problem.

Key components of optimization algorithms include useful knowledge of optimization problem, appropriate algorithmic components and appropriate combination of them in a possible way. The combination of metaheurances in relation to the four criteria, i.e. the types of algorithms selected for the composition, the level of composition, the order of algorithms to be implemented and the control strategy, is done. Also, metaheuristic algorithms can be combined with other optimization algorithms such as tree search, dynamic programming, mathematical programming, planning limitations and SMT problem solving grounds.

The banned search algorithm

It is a metacognitive approach first presented by Glover and his colleagues in 1986 [20]. This method is also known as a dynamic metaheuristic method, due to short-term or long-term memory which is its main strategy. When the search is trapped in a local minimum, this method can find another solution to eliminate this trap and make further progress, because it is the use of a concept called the list, in which movements related to the current solution are stored with the banned label, so that they are not re-selected as a new move in the interations of the prohibited law 4, after which no loops are stored and also they won't be created. In addition, the prohibited search for information about recent solutions keeps the total cost for each solution and the number of times a movement has been searched. This information can determine the status of the search process, because its large number indicates that the search process is stuck in the local optimal end and the search must end. How long a move remains in the banned list depends on the length of the list and the list is updated each time a move is added to the list. If a move marked as prohibited can adequately meet the spreading rules, it can be re-selected and removed from the list. Finally, if the prohibited search fails to find more results through the repetitions along the route, the search is finished and the optimal solution is returned. The reasons for choosing the banned search algorithm among other metaheuristic algorithms are given below:

1. The reason for its popularity is the simplicity of the scheduling process, as well as having three types of memory that prevent the search from being located locally optimally. Furthermore, it also prevents repetitive movements and solutions due to a concept called prohibited list and self-defusing criteria. Therefore, the speed of finding the best answer (global end) increases[21].

2. The prohibited search uses definitive movements that reduce diversity due to primary solutions and other parameters.

3. Instead of spending more time in areas where solutions are not desirable, the banned search is devoting more effort to exploring areas with more suitable and desirable solutions [21]

the banned pseudo-search code is listed in Algorithm 1.

The primary solution for creating the initial solution, a stochastic method

In this study, the search algorithm is used for almost the nearest neighbor.

Searching the approximation of the nearest neighbor

This study uses the nearest neighbor to create the primary solution in the banned search algorithm. In order to obtain accurate results and preventing the start of the search by solving the mistake and also to minimize the time to find the nearest neighbor, executive costs and memory consumption.

Consider the set of X objects, depending on the accuracy of the e parameter and the R distance and a data structure in which each q query object returns its nearest neighbor in X. Some other information about the X set is stored in a data structure, used to find the nearest neighbor to X without any computational operations. In the next section, the question is described as almost the nearest neighbor.

The question of finding the nearest neighbor is as follows: the algorithm 1 pseudo-code is a prohibited search code.

Algorithm 1 pseudo-code search is prohibited

//construct initial solution randomly
$S \leftarrow S_0$
sBest ← S
tabuList ← null
While (not stopping condition())
condidateList ← null
For (sCondidate in sNeighborhood)
If
(not contains TabuElements (sCondidate,tabuList))
condidateList ← condidateList+sCondidate
End
End
sCondidate ← LocateBestCondidate(condidateList)
If(fitness(sCondidate)>fitness(sBest))
sBest ← sCondidate
tabuList ← featureDifferences(sCondidate,sBest)
While(size(tabuList) > maxTabuListSize)
ExpireFeatures(tabuList)
End
End
Return(sBest)

The approximate question of the nearest neighbor is:

If the object y exists in X, so that (q.y)≤R

Then z is returned in X: (q.z)≤(1 e)R

If there is no y object in X: (q.z)≥(1 e)R

Then No is returned.

Approximately the nearest neighbor can be reduced to almost the nearest neighbor: Allow d and D to be the smallest and largest distance. Build approximately close neighboring structures for:

$$\boldsymbol{R}=\boldsymbol{d}.(\boldsymbol{1}+\boldsymbol{e})\boldsymbol{d}.(\boldsymbol{1}+\boldsymbol{e})\boldsymbol{2d}.\dots\dots..\boldsymbol{D} \quad (13)$$

To query point q, search the entire approximate structure of the nearby neighbor with:

$R=d.(1+e)d.(1+e)2d.\ldots\ldots..D$

Restore the point found in the unproductive circle with the smallest radius.

The answer is approximately the nearest neighbor of q

Problem solving the nearest neighborhood approximation with hash method:

To solve the near neighbor approximation problem, localization with hash sensitivity 1 has been used. [23] Despite many advances and different exploratory search algorithms, finding the optimal solution by each of the newest algorithms is not yet guaranteed. Therefore, improvements in algorithms and technological advances to solve these problems continue. In this study, in order to increase the efficiency of the prohibited search engine algorithm, the fruit fly optimization algorithm is used and is applied when the banned search algorithm cannot be improved further. The next part describes the optimization algorithm of fruit flies.

The Algorithm of Fruit Flies Algorithm

Algorithm optimization of fruit flies presented by Penn in 2012 [24] is based on the behavior of fruit fly search for food, whose behavior consists of two stages. Fruit flies have better smell and visual ability than other insects, so that they can detect food sources from 40 kilometers away. In the first stage, fruit flies first accidentally fly to several food sources around the group of fruit flies through Osphresis (sense of smell). This phase is called the process of exploring through odor, in the second stage, called the visual exploration phase, the fruit fly flies through its sight to the best source of food and gets closer to the food source until they stopped. It's achievable, it's repeated. As a result, during these processes, the location of the group of fruit flies is updated.

Condition of stopping in the main algorithm of the prohibited search,

The condition of stopping and the repetition of the process can be the length of the prohibited list, so that,

$size\ (tabuList)>maxTabuListSize$

Or

$f(best)\geq f(initial\ solution)$

In this study, in order to minimize the response time of the process, cost and energy consumption, the condition of stopping in the proposed model of prohibited search with approximately the nearest neighbor based on fruit fly algorithm 4, the maximum response time is considered as:

While $\boldsymbol{ResponseTime} \geq \boldsymbol{maxResponseTime}$

Then, the process stops.

The proposed ATS-FOA algorithm,

The proposed ATS-FOA algorithm that schedules the presented tasks among virtual machines in the fog nodes of the cloud-fog model, will optimize the prohibited search. According to Figure 3, this algorithm is based on a prohibited search algorithm in which the initial solution is constructed through approximately the nearest neighboring and localization with hash sensitivity. The condition for stopping the maximum response time is called to optimize fruit flies to improve the efficiency of the prohibited search, when no better solution is found by the prohibited search. In other words, fruit fly optimization starts with the list of the best candidateList solutions of the banned search algorithm and removes the old banned search solutions from the list and produces new solutions. So the list of candidates will be updated and improved. Also, those non-improvement solutions that wish to escape from the local minimum accept their fruit fly optimization. The proposed algorithm is described below in Algorithm 2:

Step 1: Create the initial solution S0 with approximately the nearest neighbor and localization with hash sensitivity and place S=S0 and sBest=S.

Step 2: Produce a sNeighborhood subset of solutions.

Step 3: Select the best sCandidate in sNeighborhood, and if the tabuList doesn't contain sCandidate, add sCandidate to candidateList.

Step 4: Otherwise, if there is the best answer on the prohibited list, end the prohibited search and execute the fruit fly search with the last prohibited list.

Step 5: Add the best fruit fly optimization solutions to candidateList.

Step 6: If Response Time > maxResponseTime

Stop the process, otherwise, go to step 2.

Algorithm 2Proposed 2 pseudo-code

```
// Finding the best initial solution using the method
approximate nearest neighbor
S←S
sBest ←S
tabuList ←null                      //Consider the prohibited list
While (not stopping condition())    //As long as the termination condition is not met
 CandidateList ←null                //Consider the candidate list

//Choose the best answer among the neighbors, if it is not in the prohibited list, add it to the candidate list
For (sCandidate in sNeighborhood)
If (not contains TabuElements (sCandidate , tabuList))
            candidateList ←candidateList + sCandiate
End

End

//Otherwise, if the best answer is in the forbidden list, terminate the forbidden search and run the fruit fly
search with the last forbidden list.

Return (candidateList)

Call FruitOptimization (candidateList)

//Then stop the process and otherwise go to step 2

While (ResponseTime > maxResponseTim)

End

Return (sBest)
```

```
//پیدا کردن بهترین جواب اولیه با استفاده ازروش approximate nearest neighbor
 S ← S0
 sBest ← S
tabuList ← null                        لیست ممنوعه را در نظر بگیر:
While (not stopping condition ())      تا زمانی که شرط خاتمه برقرار نیست
 candidateList ← null              لیست کاندید را در نظر بگیر
بهترین جواب را بین همسایگان انتخاب کن، اگر اگر در لیست ممنوعه وجود نداشت آن را به لیست کاندید اضافه کن.
 For ( sCandidate in sNeighborhood)
 If (not contains TabuElements (sCandidate,
  tabuList))
                    candidateList ← candidateList + sCandidate
 End
 End
 در غیر این صورت، اگر بهترین جواب در لیست ممنوعه وجود داشت، جست‌وجوی ممنوعه را خاتمه بده و جست‌وجوی مگس میوه را با آخرین لیست ممنوعه اجرا
 کن.
 Return (candidateList)
 Call FruitFlyOptimization (candidateList)
اگر ResponseTime  > maxResponseTime سپس فرایند را متوقف کن، در غیر این صورت، به مرحلهٔ ۲ برو.
 While (ResponseTime  > maxResponseTim)
 End
 Return (sBest)
```

Figure 3 Diagram of the proposed method

Figure 3: diagram of the proposed method

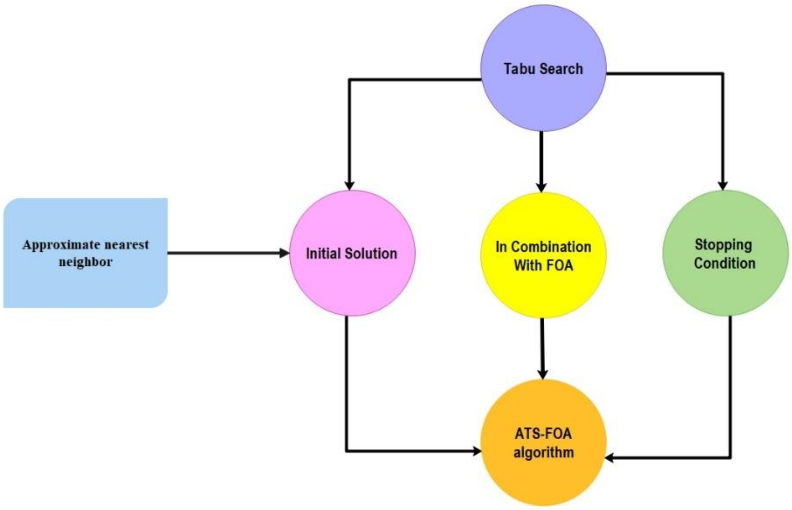
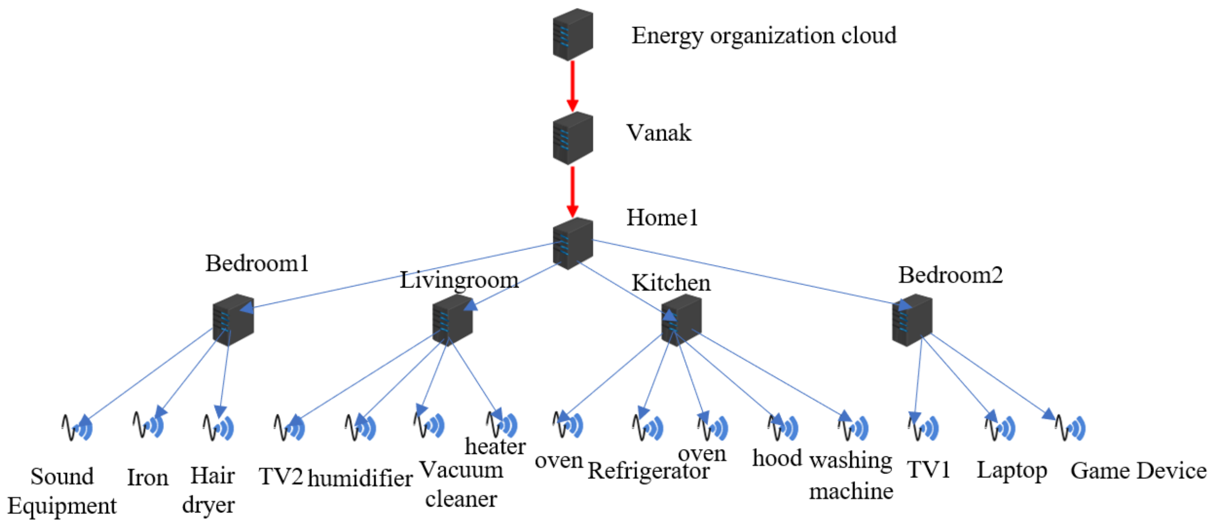


Figure 4 Configuration of the proposed model with IFogSim



Simulation

simulation settings

In this study, a smart home in Tehran has been designed to implement the proposed model and algorithm. Different parts of the home include a kitchen, bedroom and living room where smart electronic devices included in this model are set-up and are shown in tabled. 2. These smart devices, which have sensors to measure their energy consumption calculation parameters, are in the first layer of this topology. According to Figure 4, smart devices in each part of the home are connected to the fog node from the same part of the home in the second layer of topology and send their data to them. Considering the number of smart appliances in each section and the extent of home space, more than one fog node can be placed in each section, but due to the importance of cost factor, in this study, a fog node is considered in each part of the home and adjusts the parameters of fog nodes for the initial processing of internet data of objects. The parameters for fog nodes and cloud system are shown in Table 3. Also, the specifications of the hardware used in Table 4 are listed.

Table 2 Smart Electronics

Different parts of the home	**Electronic appliances**
kitchen	Refrigerator
	washing machine
	dishwasher
	Outlet hood
	electric oven
Bedroom 1	Television
	Game Device
	Laptop and mobile chargers
Bedroom 2	clothes iron
	sound equipment
	Hairdryer
living room	Television
	vacuum cleaner
	humidifier
	electric heater

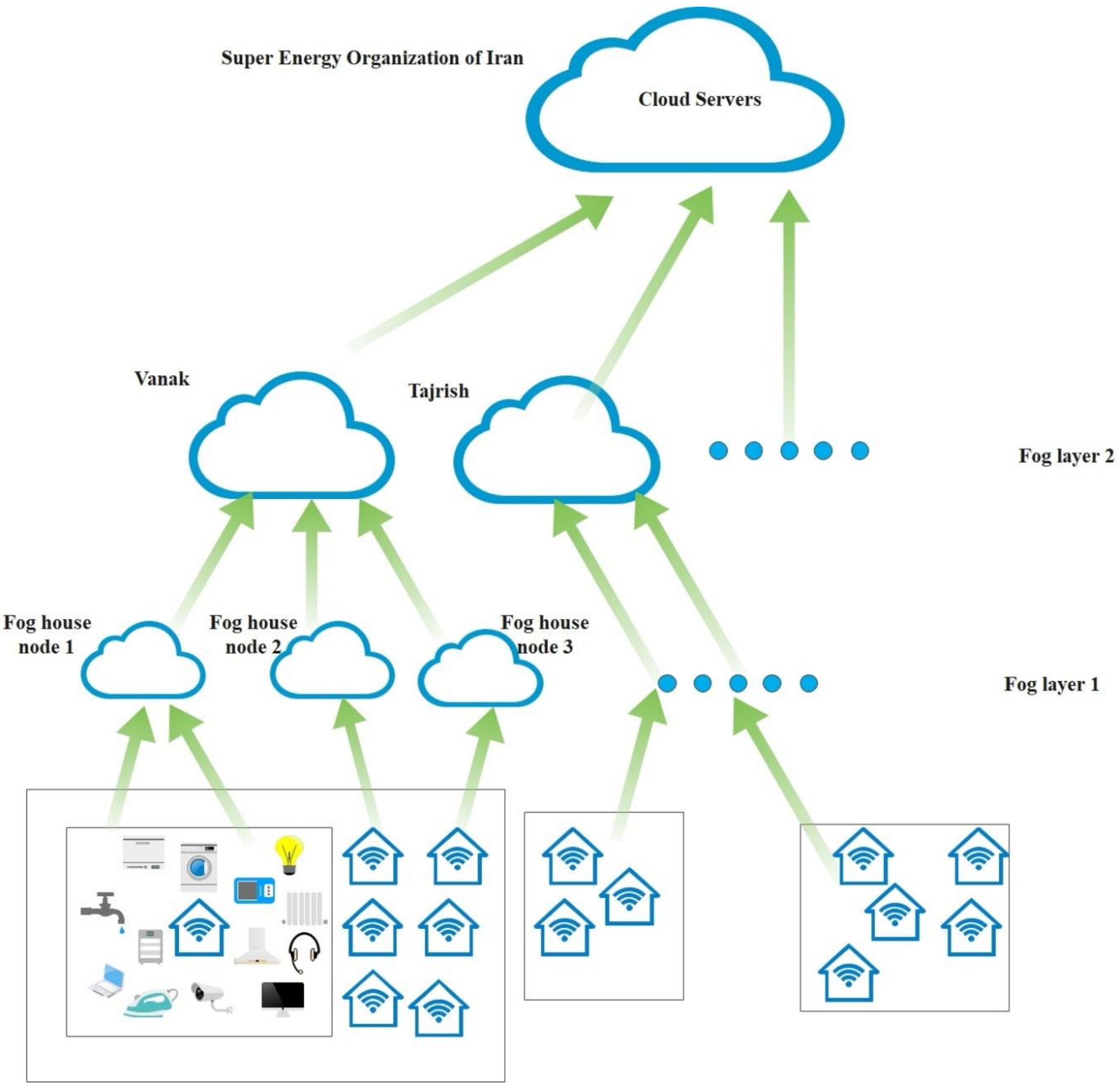
Table 3 Fog Node Settings

parameters	Values
Levels	0-3
Uplink BW	10000
Downlink BW	270-10000
Mips	2000
Ram	4000
Rate / MIPS	0.01

Table 4: The infrastructure specifications used to implement

parameters	Values
System Architecture	X86
Operating system	Linux
Vmm[1]	Xen
CostPerMemory	0.05
CostPerStorage	0.001
CostPerBw	0.0

Figure 5 of the infrastructure for the smart home with three layers of fog of a real data set



[34] from a smart home in this simulated architecture and for evaluating the proposed algorithm and comparing the results obtained with the banned search algorithm and also investigating different hypotheses to determine the number of fog layers between the cloud and internet devices of objects have been used in the first assumption. The three-layer model has fog among the cloud system and smart appliances. In these three layers, the number of nodes is fog 7 and the results of this assumption are shown in Table 5. Then, in the second scenario, it is assumed that the second layer of Fog will be removed from this architecture and the fog nodes of different parts of the home will be attached to the node of their geographical area (as an example of one region in the assumed model) that the number of fog nodes in this assumption is 6. On the other hand, in the third scenario, an additional layer of fog is added to this architecture, which is assumed to be a fog node for the residential complex. Therefore, it increases the number of fog nodes by 8. Then, these assumptions were evaluated in a simulated model simulated by Clodsim tool for scheduling tasks between fog nodes and virtual machines in a fog node using ATS-FOA algorithm. The emulsor used is Claude Sim. Claude Sim simulator supports the behavior modeling of system elements,

such as data centers, virtual machines, and resource allocation policies. Therefore, it can be used to model the computational cloud environment and test the efficiency of the proposed algorithms. This emulation is based on the Java language and the event, i.e., the ability in which it is defined.

They communicate with each other by sending events. In this simulation, the number of tasks sent for processing in different scenarios with layers 2, 3, 4, 400 is considered. Each task also has 103 instructions. Also, in the case assumption, the coefficients of the runtime and memory variables used in the cost function are determined by the decision-taker:

$$\mathbf{0.9}\ \boldsymbol{CPU\_Execution\_Time} + \mathbf{0.1}\ \boldsymbol{Allocated\_Memory} \quad (25$$

All experiments with 20 seconds project profile (virtual machines execution time is 20 seconds) have been carried out and the results of Table 5 have been obtained.

Results of Simulation

The aim of this evaluation and simulation is the implementation of two banned search algorithms and ATS-FOA under the same conditions in the proposed infrastructure and obtaining the response time values of 400 assigned tasks, execution time, memory consumed for processing. These tasks and the cost of implementing them are based on the time of implementation and memory consumption in order to compare the two algorithms in the same conditions and show the efficiency of the new algorithm. As obtained from figure 8, response time.

The requested works (400 tasks) in the proposed infrastructure with three layers of fog, with a new scheduling algorithm of 2744 in 20 seconds project profile, is 2744.07 seconds (meaning that with 2744.07 seconds) The fog 7 node of the response time of 400 tasks is obtained from a division of 2744.07 to 20, and so on for all values (and by performing the banned search algorithm is 2954.28 seconds). The reason for the decrease in response time in the new algorithm is that the prohibited search. In order to allocate tasks to virtual machines in fog nodes with random solutions, the allocation of tasks to busy virtual machines may increase the waiting time of tasks in the queue and their response time. While in the new method, for the reason of choosing the most appropriate answer as the initial answer and indeed appropriate allocation of tasks to virtual machines that have less processing load, speed of operation and consequently, The response time will be lower.

Moreover, in the new algorithm, according to algorithm 2, if the best neighbor answer is on the prohibited list and its breathing criterion is better than the best answer found so far, the algorithm will not move to it and the search is finished, immediately the fruit fly algorithm with the prohibited search candidate list will start searching.

Consequently, searching is not trapped in the local optimal and the best improvement is achieved through the fruit fly search. With the best answer, the tasks assigned to the most suitable virtual machines and are executed in a shorter time. In the prohibited search, despite the prohibited list, it is prevented from falling into the local trap, but due to the condition of breathing, the answer, if any, is on the prohibited list, so it may be trapped in optimal local conditions.

Furthermore, due to the fact that the stop condition in the new algorithm is the maximum response time considered, the response time of the entire search process will not exceed one limit. Figure 9 shows the runtime of the processors to process 400 tasks by implementing two methods. Similar to response time, running time for the new algorithm will be less than prohibited search, because by assigning tasks to virtual machines that have less processing load, it causes things to run in less time. Also, failure to calculate the breathing criteria for the answers and not repeating them in the search process reduces the time of performing the tasks. Figure 6 (shows the memory used to process tasks by implementing two algorithms.

As we can see, the amount of memory consumed by the new algorithm has decreased by a very small amount compared to the prohibited search. This is because after searching almost the nearest neighbor with hash method to find the initial answer, the memory assigned to it is released.

Table 5 - Comparison of two algorithms based on the values of objective factors in 3 scenarios is considered

400 tasks	**execution time**			**response time**			**Consumable memory**			**Cost**		
scenario	1	2	3	1	2	3	1	2	3	1	2	3
ATS-FAO	2677,38	2743,93	2962,29	2677,76	2744,07	2962,44	3001104	3210224	3350480	0,08694	9.21E-02	9.62E-02
Tabu Search	2800,01	2954,13	3069,24	2800,17	2954,28	3069,38	3001304	3267448	3485424	8.61E-02	9.38E-02	.100101

In addition to this, in the new algorithm, due to lack of calculation of breathing criteria, keepers are not assigned to it. Also, the only recently observed answers are kept as short-term memory, and other information such as movements made or the number of times a prohibited movement or route is searched, which is stored in a prohibited search and memory is assigned to them. Figure 7 (the cost of execution and processing of works and in fact, the efficiency based on the time of execution and memory consumed by the two algorithms shows that the calculated cost for the new algorithm is less than the prohibited search. Figure 10 also shows the comparison between the results of two algorithms for the three scenarios, as it is suggested, if the number of fog layers is lower, the values of all the target factors are also better for the new algorithm, will increase. The number of fog nodes and consequently increasing storage and processing capacity not only increase the speed and efficiency in this system, but also increases the time and cost.

Suitable and cost-effective number should be determined in each field and for each system depending on the type of activity. The convergence of an algorithm is the same number of rounds that the algorithm takes to achieve the optimal answer. Therefore, the convergence of the proposed algorithm in simulation to achieve the best scheduling has been obtained. For example, in the first scenario for 400 tasks sent to the system with two layers of fog nodes and six fog nodes, the number of rounds to obtain the contingency timing of fog nodes is 29 rounds.

Figure 6 - Memory consumed by processing 400 works in smart home infrastructure with three layers of fog

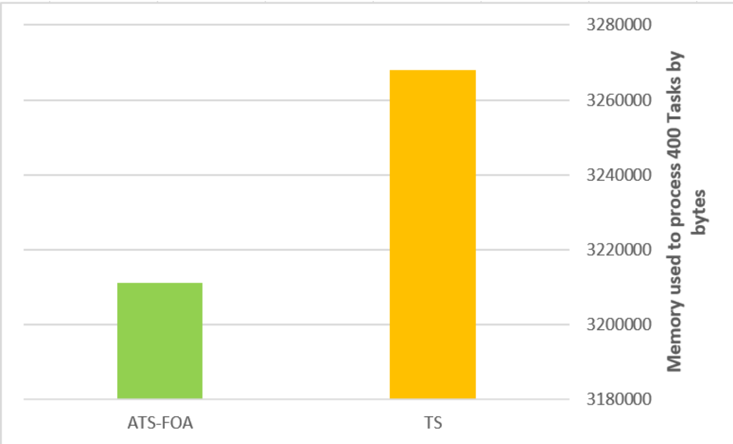


Figure 7 - Calculated cost to process 400 tasks on the proposed platform with three layers of fog

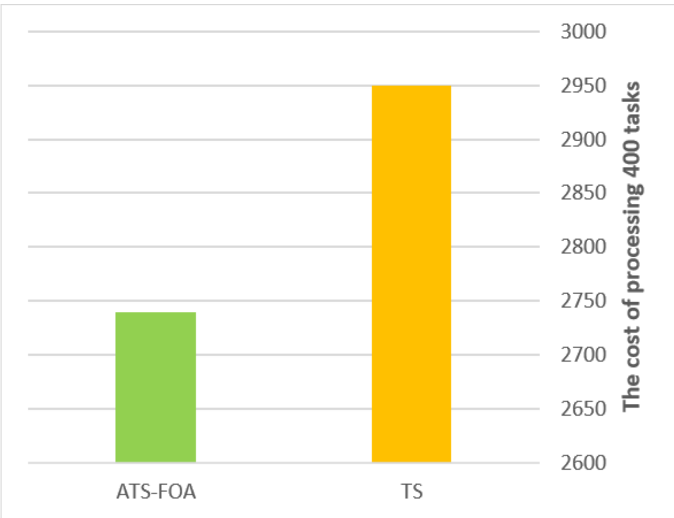


Figure 8 - Response time of 400 tasks on the proposed platform with three layers of fog for smart home

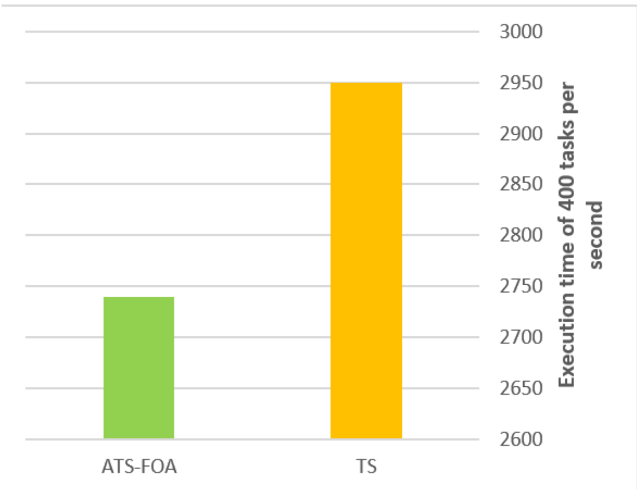
The algorithm starts to run from [4, 6, 1, 2, 3, 5] for the scheduling of fog nodes, and after 29 repetitions it reaches [2, 3, 5, 6, 1, 4] the order of the nodes of fog. Best Selected Schedule Fog Nodes: [2, 3, 5, 6, 1, 4]

The time complexity of both algorithms is also calculated as follows:

Time Complexity of Tabu Search: $(n)=1+n3+n2 \in O(n3)$

Time Complexity of ATS-FOA: $(n)=logn+n3+n \in O(n3)$

Figure 9 - The execution time of 400 requested tasks on the proposed platform with three layers of fog

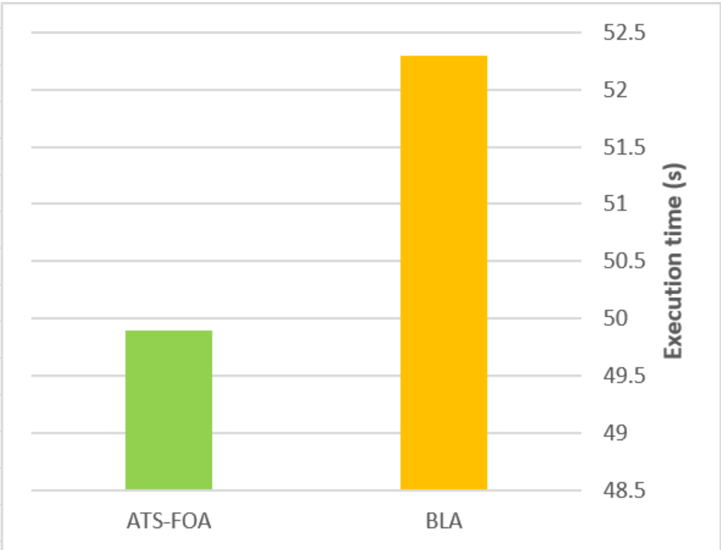


Therefore, the temporal complexity of both the banned search algorithm and the proposed algorithm is from the 3n order, but according to the results of the experiments, the time of implementation of the proposed algorithm is less than the prohibited search. On the other hand, the time function related to the proposed algorithm is smaller than the time function of the prohibited search algorithm:

Logn n3 n ≤ 1 n3 n2 n logn≤n2

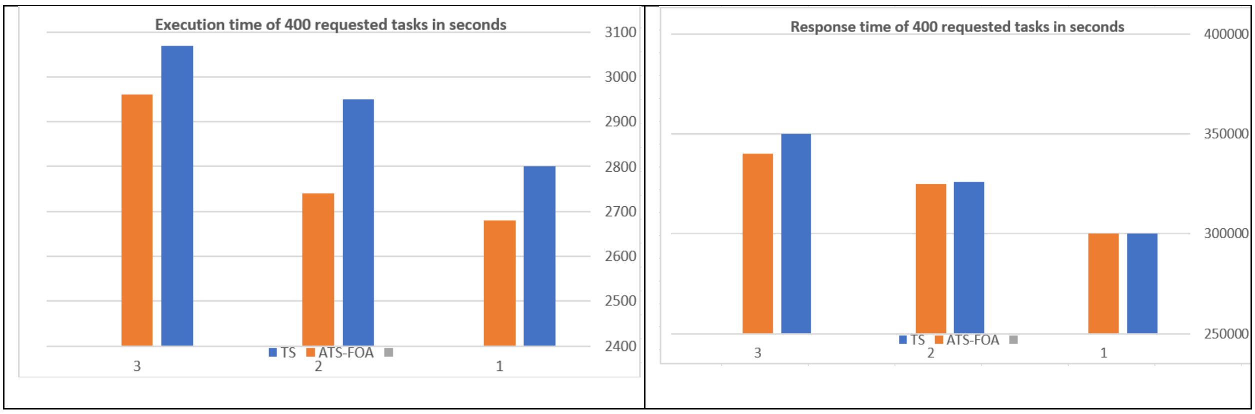
For validation of the proposed algorithm in this study, its results are compared with the results of the proposed algorithm [7] under some conditions. In article 7[7], five tasks are submitted to the system and each task consists of five tasks and each task consists of 10 instructions, therefore, the proposed algorithm. This article has been simulated and implemented with the same conditions with 20 nodes of Fog and the results of comparing it with the BLA algorithm (Bees Life Algorithm) article in Figure 11 (Figure 10 shown the results of comparing the BLA algorithm, ATS_FOA

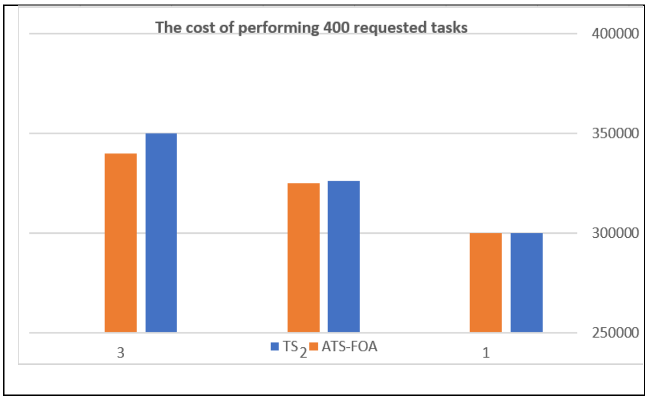
Figure 10 -Comparison results of BLA,ATS_FOA algorithm

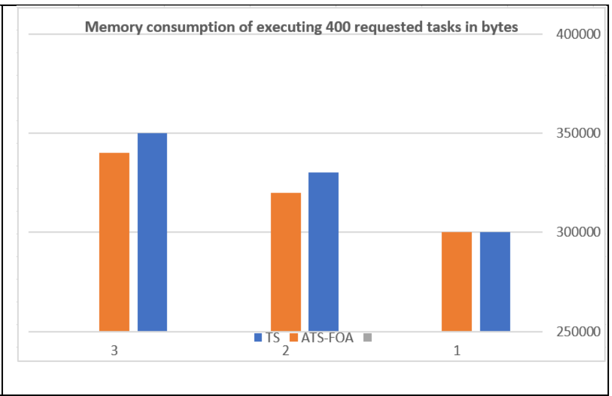


As obtained from figure 11, the total execution time (25 tasks) with 910 instructions in the infrastructure with 20 nodes of fog by ATS-FOA algorithm is 49.87 seconds, while the implementation of this number of tasks by BLA algorithm is 52.42 seconds. Therefore, the proposed ATS-FOA algorithm had better performance than BLA algorithm.

figure 11, the total execution time (25 tasks) with 910 instructions in the infrastructure with 20 nodes of fog by ATS-FOA algorithm is 49.87 seconds







Conclusion

The development of communications and technology is increasing the volume of produced data, which takes a lot of time and cost to process this volume of data. On the other hand, technologies are growing into systems with high speed, delay and lowest cost. Consequently, creating systems that can process this volume of data in the shortest possible time and at a low cost is much needed. In order to create these systems, infrastructures should be provided that can support these goals. In this study, an infrastructure with four elements based on cloud and fog is proposed and evaluated using a case study on smart homes. Also, by implementing this platform for smart home with real data, the scheduling problem of requested tasks among virtual machines and among the fog nodes of this platform has also been done using a new proposed method based on prohibited search. So that the cost, response time, running time and memory consumed by implementing different scenarios have been obtained. According to the obtained results, by adding more fog nodes and layers and consequently, more processing and storage capacity increases the infrastructure of systems, cost and time. The reason for this increase is that by adding more layers between the cloud server and smart devices, the delay of transmission in the network increases and as a result, the response time increases. The cost and memory consumption also increases with increasing storage capacities. Therefore, considering the type of system and its application, a cost-effective and high-efficiency infrastructure should be designed. This proposed method can be used for other smart city systems. In future tasks, the method can be improved with the aim of lower consumption memory. Also, this infrastructure and proposed method can be used for the banking system of the country and other organizations.

References:

1. B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions", Future Generation Computer Systems, Vol. 79, 2018, pp. 849–861.
2. M. Amadeo, A. Giordano, C. Mastroianni and A. Molinaro, "On the Integration of Information Centric Networking and Fog Computing for Smart Home Services", Springer, Cham, 2019, pp. 75–93.

3. M. Marjani, F. Nasaruddin, A. Gani, A. Karim and I. Abaker, "Big IoT Data Analytics : Architecture , Opportunities , and Open Research Challenges", Vol. 3536, No. c, 2017, pp. 1–17.
4. A. Mebrek, L. Merghem-boulahia and M. Esseghir, "Efficient Green Solution for a Balanced Energy Consumption and Delay in the IoT-Fog-Cloud Computing", 2017. [5] S. Sarkar, S. Chatterjee and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things", IEEE Transactions on Cloud Computing, Vol. 6, No. 1, 2018, pp. 46–59.
5. A.N. Toosi, R. Mahmud, Q. Chi and R. Buyya, "Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds", in Fog and Edge Computing, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2019, pp. 79–101.
6. S. Bitam, S. Zeadally and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm", Enterprise Information Systems, Vol. 12, No. 4, 2018, pp. 373–397. [8] A. Yassine, S. Singh, M.S. Hossain and G. Muhammad, "IoT big data analytics for smart homes with fog and cloud computing", Future Generation Computer Systems, Vol. 91, 2019, pp. 563–573.
7. S.A.A. Naqvi, N. Javaid, H. Butt, M.B. Kamal, A. Hamza and M. Kashif, "Metaheuristic Optimization Technique for Load Balancing in Cloud-Fog Environment Integrated with Smart Grid", Springer, Cham, 2019, pp. 700–711.
8. L. Peng, A.R. Dhaini and P.-H. Ho, "Toward integrated Cloud–Fog networks for efficient IoT provisioning: Key challenges and solutions", Future Generation Computer Systems, Vol. 88, 2018, pp. 606–613.
9. P.G.V. Naranjo, Z. Pooranian, M. Shojafar, M. Conti and R. Buyya, "FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments", Journal of Parallel and Distributed Computing, Jul. 2018.
10. S.K. Sood and K.D. Singh, "SNA Based Resource Optimization in Optical Network using Fog and Cloud Computing", Optical Switching and Networking, Dec. 2017.
11. S. Singh and A. Yassine, "IoT Big Data Analytics with Fog Computing for Homehold Energy Management in Smart Grids", Springer, Cham, 2019, pp. 13–22. [14] Q. Li, L. Zhao, J. Gao, H. Liang, L. Zhao and X. Tang, "SMDP-Based Coordinated Virtual Machine Allocations in Cloud-Fog Computing Systems", IEEE Internet of Things Journal, Vol. 5, No. 3, 2018, pp. 1977–1988.
12. Z. Zeng, X. Yu, K. He and Z. Fu, "Adaptive Tabu search and variable neighborhood descent for packing unequal circles into a square", Applied Soft Computing, Vol. 65, 2018, pp. 196–213.
13. Magazine, "Memory-based artificial bee colony algorithm for solving dynamic optimization problems", [Majid Mohammadpour and Hamid Parvin Modeling in Engineering, Volume 15, No. 51, Winter 2018, Page 113-132.

14. Development of a branch and bound algorithm for solving the scheduling problem in the flow generation system", [Seyed Mohammad Hassan Hosseini and Aliakbar Hosni, Modeling Magazine in Engineering, Dora, Assembly Workshop 15, No. 51, Winter 2017, Page 2006 - 2019.
15. Modeling magazine in "Optimal Distribution of Reactive Power Using Corn Category Optimization Algorithm", [Hossein Sharifzadeh and Nima Amjady Engineering, Volume 4, No. 18, Winter 2009, Page 67 - 73
16. G.R. Raidl, J. Puchinger and C. Blum, Metaheuristic Hybrids.
17. F. Glover, "Tabu Search—Part I", ORSA Journal on Computing, Vol. 1, No. 3, 1989, pp. 190–206.
18. H. Pirim, E. Bayraktar and B. Eksioglu, "Tabu Search: A Comparative Study", Tabu Search, September 2008.
19. F. Glover, "Tabu search fundamentals and uses", Vasa, 1995.
20. T-61.5060 Algorithmic methods for data mining Slide set 5: locality-sensitive hashing.
21. W.-T. Pan, "A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example", Knowledge-based Systems, Vol. 26, 2012, pp. 69–74.