

# New Keys for Cloud Resource Provisioning Optimization Method in Multi-Tier Style

Omer K. Jasim Mohammad<sup>1,\*</sup>

<sup>1</sup>University of Fallujah, Fallujah, Anbar, Iraq

Email: [omerk.jasim@uofallujah.edu.iq](mailto:omerk.jasim@uofallujah.edu.iq)

## Abstract

Resource provisioning is regarded as a crucial technology in the cloud-computing environment. Nonetheless, the primary challenge associated with the cloud involves ensuring resource availability while improving throughput, balancing loads, and optimizing execution time. There are two types of provisioning methodologies in a cloud environment: single-tier and multi-tier. This paper presents a novel method that combines hybrid metaheuristic optimization techniques, specifically Ant Colony Optimization (ACO) and Firefly Algorithm (FA), referred to as ACOF. This study presented an implementation of dynamic resource provisioning in a multi-tier cloud architecture. The results obtained from the proposed method demonstrate an enhancement in resource provisioning compared to other studies. Indeed, the ACOF algorithm demonstrates a reduced execution time for resource provisioning compared to alternative algorithms. Furthermore, ACOF algorithms have the potential to decrease implementation time by up to 13.2% in comparison to the execution time of alternative methods.

Received: July 05, 2024 Revised: October 06, 2024 Accepted: December 29, 2024

**Keywords:** Cloud computing; Meta-heuristic algorithms; Resource provisioning; Throughput; Resource management; Execution time

## 1. Introduction

The advancement of information technology and the rising utilization of distant services and web technologies are transforming computing through a novel Internet paradigm known as Cloud computing. This strategy involves remote data processing and computation conducted on a network of data centers managed by a third party. The technology is not novel and has been utilized for an extended period over the Internet. In this scenario, an appealing platform is utilized to host Web applications and services, including emails, Web servers, and storage, in conjunction with the requisite standards and protocols. The technology has evolved and has lately been designated as "*Cloud*". It now denotes the hardware, system software, and applications provided as services via the Internet [1].

Cloud computing optimally utilizes various distributed resources to deliver services and/or resources to clients remotely, ensuring maximum performance in minimal time. Users together utilize processing power, storage capacity, bandwidth, memory, software, and expenses. Users utilize only what is necessary at each moment, hence minimizing expenses [2]. Quality of Service (QoS) is a motivating factor for users to utilize the Cloud, as it should ensure excellent service availability on demand. Nevertheless, customers may perceive that the Service Level Agreements (SLAs) upheld by the Cloud providers are inadequate to ensure the prerequisites for operating production applications in the Cloud [4, 5].

The lack of a data migration likelihood is still another significant disadvantage of cloud computing. Customers may want to move their data and services from one provider to another, but cloud computing does not allow for this possibility [6]. The term "cloud computing" refers to a phenomenon in which a significant quantity of computing resources are dispersed across multiple data centers and can be accessed as a service through user interfaces. When it comes to cloud computing, optimizing the provisioning of resources is an extremely important

topic. In most cases, this is accomplished by selecting and delivering resources within a Virtual Machine (VM) that is hosted on a single Physical Machine (PM), with the goal of guaranteeing that applications provide guaranteed performance [7].

This paper presents a novel approach that combines hybrid metaheuristic optimization techniques, specifically integrating Ant Colony Optimization (ACO) [8] with the Firefly Algorithm (FA) [9], referred to as ACOF. This study presented an implementation for the dynamic provisioning of resources in a multi-tier cloud architecture. The results obtained from the proposed method demonstrate an enhancement in resource provisioning when compared to other studies.

Finally, this paper is organized into several sections: Section 2 delineates the boundaries of the study presented in the paper. Section 3 provides a detailed explanation of cloud resource provisioning, while Section 4 elucidates the cloud multi-tier architecture. The methods proposed have been illustrated in Section 5. The results obtained from the experiments are presented in Section 6. Section 7 provides a detailed discussion on cloud performance. In conclusion, Section 8 encapsulates the results and outlines potential avenues for future exploration.

## **2. Study Boundaries**

This part lays down the problem statement, motivations, and methodology.

### **A. Problem Description**

According to our survey, the primary obstacles of resource provisioning in cloud computing are portability, cost and execution time, and cloud architectural structure. Cloud architecture must possess the capability to transfer virtual machines between different cloud computing environments. While, the cost and execution time are crucial for the consumer, as the amount spent should correspond to the time used to fulfill their request. The cloud architecture of an application or service must be addressed while planning the provisioning scheme and developing a deployment strategy in the cloud environment.

### **B. Study Motivation**

Despite the attributes of cloud computing, numerous hurdles hinder the adoption of the technology, with the primary issues being security, resource provisioning, and allocation. The primary issues in resource provisioning are the selection of suitable resources and the provision of an adequate architecture. The optimal resolution for these two difficulties is the algorithmic selection of resources. This method employs optimization techniques, particularly meta-heuristic optimization methods inspired by natural events. Our survey indicates that meta-heuristic algorithms have been utilized only in prior studies for resource allocation in Single-tier Clouds, but not in Multi-tier architectures.

### **C. Methodology**

The technique of this study adheres to these steps:

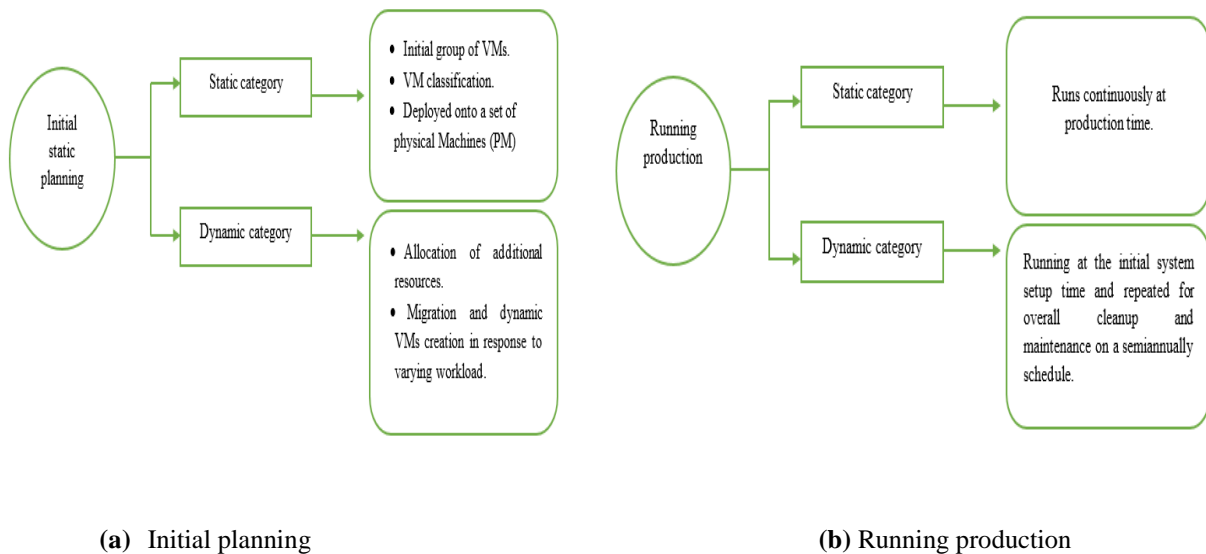
- 1 Examining the contemporary progress in Cloud computing and the administration of resources inside Cloud systems through theoretical analysis.
- 2 Analyzing and evaluating the efficacy of notable cloud provisioning resource methodologies, encompassing single-tier and multi-tier cloud architectures.
- 3 Constructing the virtual cloud infrastructure using Xen (5.6 XCP) server.
- 4 Altering the Data Center Broker to facilitate various resource provisioning techniques.
- 5 Implementing the ACOF heuristic optimization methods in multi-tier cloud environments.
- 6 Assessing and contemplating the obtained results.

## **3. Cloud Resource Provisioning (CRP)**

In cloud, computing, substantial computational resources are disseminated among data centers, accessible as services via user interfaces. An essential issue in cloud computing is the optimization of resource allocation. This is often accomplished by allocating and supplying resources within a Virtual Machine (VM) on a singular Physical Machine (PM) to ensure consistent program performance. Resource provisioning can adhere to two primary strategies. These are the reservation plan and the on-demand plan [9]. In on-demand scenarios, the consumers can access resources in the time when they needed. While, the reservation plan is a scheme where the resources could be reserved earlier. CRP is one of the major pitfalls in cloud computing and it is related to optimizing allocation and provisioning of resources. Usually, cloud computing manages resources with two functions: assigning resources and releasing a resource [10]. Regarding the relation between provisioning and infrastructure, each infrastructure provides a different set of fully requests that includes "Operating System (OS), applications, and data". Each request offered by provider or cloud users.

Various algorithms have been employed to optimize resource provisioning in cloud computing, categorized into two types: meta-heuristics and heuristics. Simulated Annealing (SA), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) are well-known examples of meta-heuristic algorithms [11]. However, constructive algorithms, greedy methods, hill-climbing, knapsack problem are well known of heuristic trade-offs and techniques [12].

CRP is often attained through two categories: static resource provisioning and dynamic resource provisioning [10]. The static category designated for applications with known demands, entirely reliant on contracts between cloud providers and end users. The dynamic category is employed when applications require fluctuating resources over time, allowing the cloud provider to allocate additional resources as necessary and retract them when they are no longer needed [12]. Two categories adhere to two fundamental processes for organizing work: "initial static planning" and "running production." Each step encompasses a specific process for both static and dynamic categories during the deployment phase. Figure 1 illustrates the operational mechanism for each phase in detail.



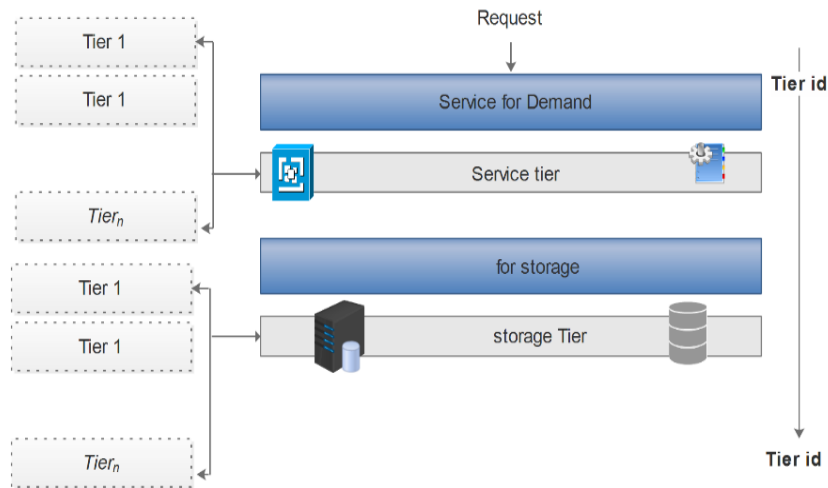
**Figure 1.** Operational mechanism for static and dynamic categories

The aforementioned figure indicates that resource provisioning in the Cloud is typically guided by performance forecasts; thus, precisely characterizing workload variations is essential for effective resource allocation. Furthermore, to attain an optimal implementation process, a series of parameters must be considered for effective provisioning, including response time, cost minimization, revenue maximization, fault tolerance, reduced service level agreement (SLA) violations, and diminished power consumption. For further details, see [13].

#### 4. Cloud Multi-Tier Architecture

The execution of tasks in cloud computing fundamentally relies on the notion of fragmentation, whereby a given work is typically divided into multiple sub-tasks. Consequently, parallel processing is an optimal choice for task execution, and efforts should be made to distribute resources to these activities. Additionally, orchestrating and scheduling the execution of jobs on the Cloud. Single-tier and multi-tier architectures are two fundamental and recognized structures that rely on resource allocation in a cloud-computing environment. Single-tier architecture comprises a set of servers that deliver resources by processing requests from cloud users, retrieving information from the application server, and storing data in the database, see figure 2. No interactivity is permitted, nor is sharing between servers allowed in this architectural framework.

Moreover, the multi-tier cloud architecture divides an application into several tiers, and both the cloud user and server components can be further segmented into multiple sub-applications. The classification completely segregates the application into distinct tiers based on their functionalities. Furthermore, servers within the service or storage tier may be categorized into distinct sub-tiers. Typically, each application possesses a defined set of requirements and limitations articulated by the application owner through a Service Level Agreement (SLA). Figure 2 illustrates the architecture of a multi-tier application.



**Figure 2.** Architecture of a multi-tier style

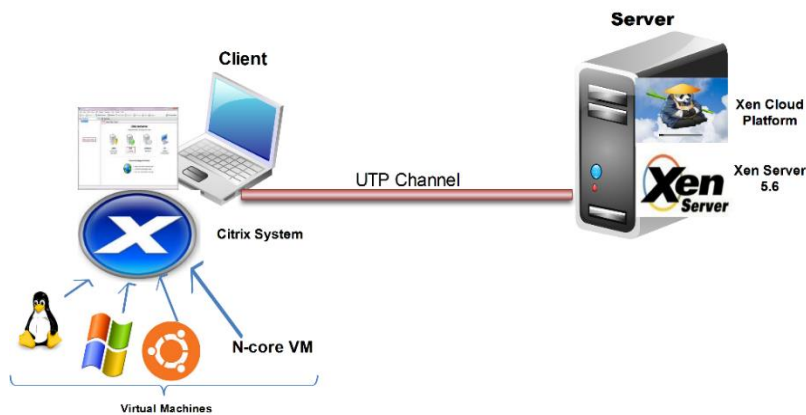
This architecture is typically assessed through performance demand, which is characterized by the maximum end-to-end response time for a request and the associated deployment cost. This duration can represent either a mean response time or a high percentile within the response time distribution. The cost represents the financial limitation associated with the overall deployment of the application. Furthermore, each tier is associated with a distinct resource relative to the maximum number of servers within that tier, as illustrated in figure (2). The multi-tier architecture ensures the modular structure of cloud applications and enables effective management of their tiers. The server set of an application can be categorized into several subsets, with each subset comprising servers that belong to the same tier.

## 5. Proposed Method

This section presents our proposed methods for dynamic resource provisioning using Ant Colony Optimization (ACO) and Firefly Algorithm (FA) algorithms in Multi-tier Clouds with goal minimizing total execution time and cost.

### 5.1 Experimental Environment

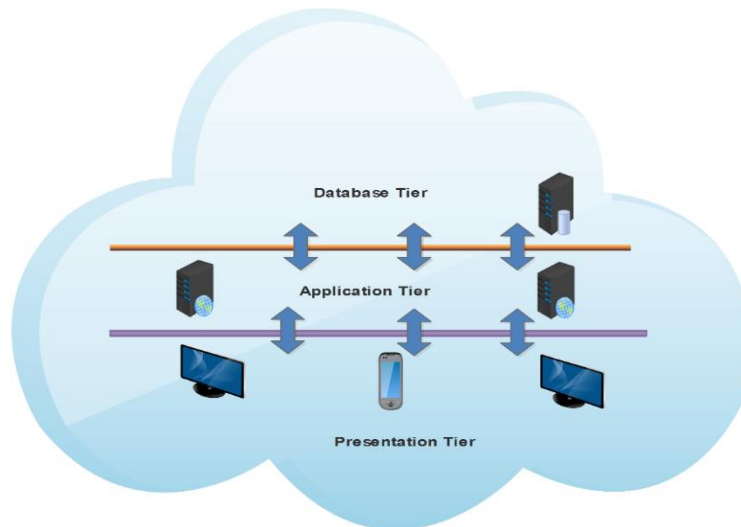
The experimental setup comprises a cloud network and a single-processor local workstation. The cloud network comprises the Xen (5.6 XCP) server and the client utilizing Citrix as the virtualization solution with N para-virtual machines. The cloud server employs a Core i5 (4.8GHz) processor, 8GB of RAM, and a 500GB HDD as its primary hardware, whereas the client computer is equipped with a Core i3 (2.4GHz) processor, 2GB of RAM, and a 300GB HDD, as illustrated in the figure 3.



**Figure 3.** Cloud Networking

In multi-tier architecture, requests go through various tiers, where resources are released at each layer, allowing them to be utilized by subsequent requests and/or tiers, thereby conserving time, minimizing resource utilization, and reducing costs.

As shown in figure 4, cloud computing used three tier architecture named as presentation tier, Web server tier, and Application server tier. All of tiers have specific job and have specific mechanism for instance arrange the request process, passing request to another application server, send the response to the client, passing the processed information to database.



**Figure 4.** Main multi-tier cloud architecture layers

ACOF is a dynamic resource provisioning method that integrates Ant Colony Optimization (ACO) and the Firefly Algorithm (FA) inside a multi-tier cloud architecture. Each tier offers certain features. This architecture offers significant scalability and reliability. Nonetheless, the allocation of resources among these layers will be more challenging due to their interdependence. Generally, ACFO exploits firefly algorithm to compute the fitness function, update tasks distribution, and rearrange the task location, allocation resources as task completion need. Finally, this study adopted of fitness value that represented Expected Completion Time (ECT) of requests and calculate Estimated Execution Time (EET) as follows:-

$$EET = (CT - ST) + DU \quad \dots (1)$$

$$ECT = ST + DU + EET \quad \dots (2)$$

Where:

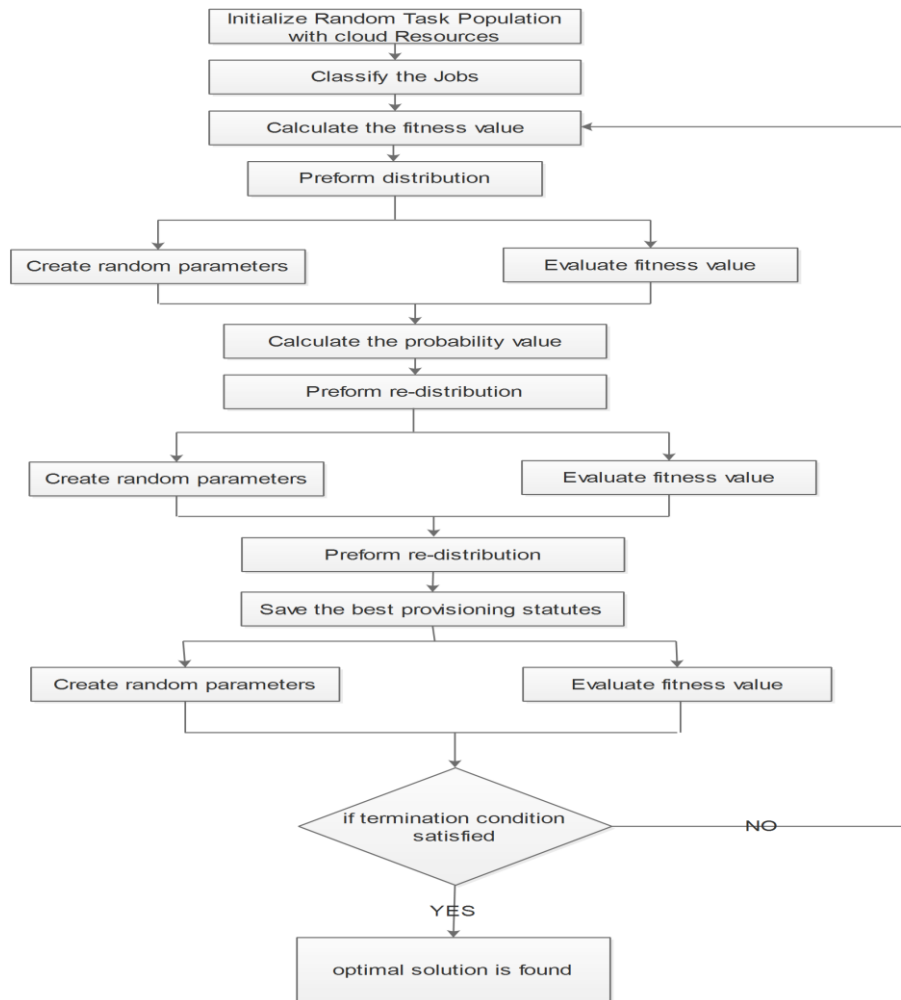
**CT:** completion time and **ST:** Start time of requests.

**DU** acts the duration time between arrival requests until to start. In addition, this study used **ECT** to select local best position ( $L_{best}$ .)

$L_{best}$ : represents best resource in significant groups.

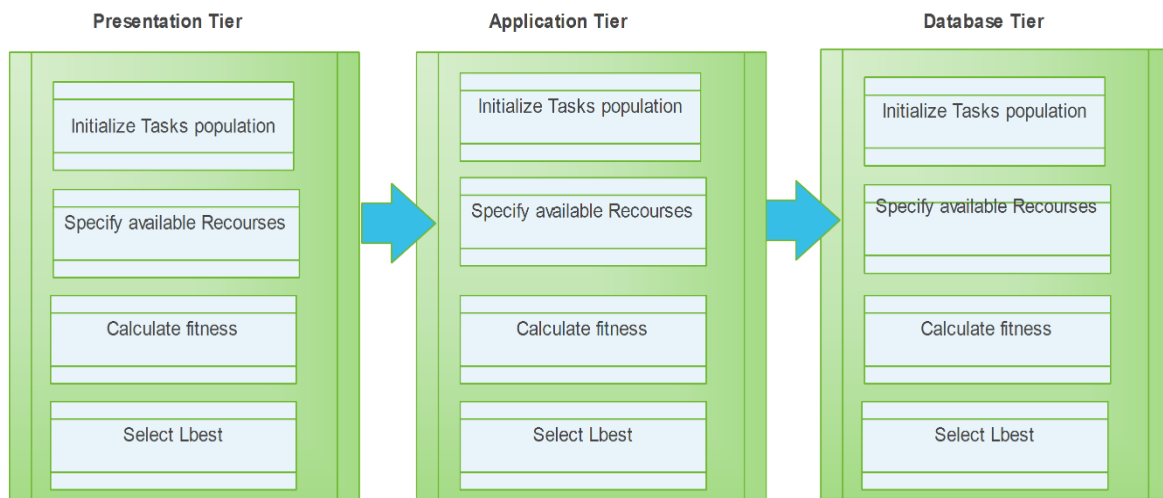
## 5.2 ACOF Implementation

This section outlines the primary step of ACOF implementation within a multi-tier cloud environment, focusing on dynamic resource provisioning. ACOF facilitates the identification and provision of optimal resources within each tier of Multi-tier Clouds by conducting local searches to determine the best resource for each tier of the request, ultimately selecting the most suitable resources from the available options. The flowchart of the ACOF algorithm for optimizing resource provisioning is illustrated in Figure 5. The workflow of the ACOF algorithm involves modulating the optimal solution from various tasks.



**Figure 5.** ACOF flowchart

To apply the ACOF algorithm, there are steps should be taken and repeated in each tier mentioned in figure 4. Figure 6 shows the main steps in each tier.



**Figure 6.** ACOF in Multi-tier clouds

While, the algorithm of ACFO listed as in Algorithm 1.

**Algorithm 1. ACFO Algorithm**

```

1.  Generate Initial Population (tires, tasks)
2.  Set parameter for job (j) and cloud tiers
3.  Set t=0;
4.  For each Request i = 1, ... ,P
5.  For each tier t = 1, ... ,Tiers
6.  For each resources j = 1, ... ,R
7.  Calculate EET from Eq. 1
8.  Calculate ECT from Eq. 2
9.  If (ECT[i][j] < ECT [i][j - 1])
10. (Lbest[i][j] = ECT [i][j]);
11. Else
12. {
13. If (Lbest[i][j] > Lbest[i][j - 1])
14. Temp = Lbest[i][j - 1]);
15. Lbest[i][j - 1] = Lbest[i][j];
16. Lbest[i][j] = Temp);
17. End if
18. If Lbest needs update and migration
19. {
20. Finding out migration source machine
21. Finding out migration object machine
22. Finding out migration different value
23. Finding out migration VM
24. Migrate
25. Else
26. Output finalLbest.
27. End if
28. End if
29. End for
30. End for
31. End for

```

In ACOF algorithm, following steps should be achieved:-

- Generate initial population, and then calculate EET to calculated ECT (fitness value) for each resources in each tier of the job.
- For each tire compute **L<sub>best</sub>** .
- Select **L<sub>bets</sub>** resource based on ECT and depending on **L<sub>bets</sub>** resource.
- Check the value of **L<sub>bets</sub>** if it needs for migration.
- Achieve the migration.
- **L<sub>best</sub>** resource is the resource that could be used in resource provisioning because it consumes less execution time if compared with other available resources.

In the initialization step of the ACOF algorithm, Initial population and cloud tier are initialized. The *t* is assigned to 0 values in order to avoid the infinite process, while, the *i* , *j* value are 1. In general, the ACOF algorithm depends on one outer loop [For] (line 4) and two inner loops [*t*], and [*j*] (lines 5, and 6), the outer loop indexed by *p* (the length parameter)]. While, the inner one (variation loop) that used to assign an element to each tier in the obtained resources. The operation of ACOF algorithm terminates when *Lbest* equals to *ECT*. The time complexity of the algorithm in the line 1 takes constant time (*n*), while in line 2 needs  $O(n)^3$ . Therefore, the total cost of ACOF algorithm is  $O(n)^3$ .

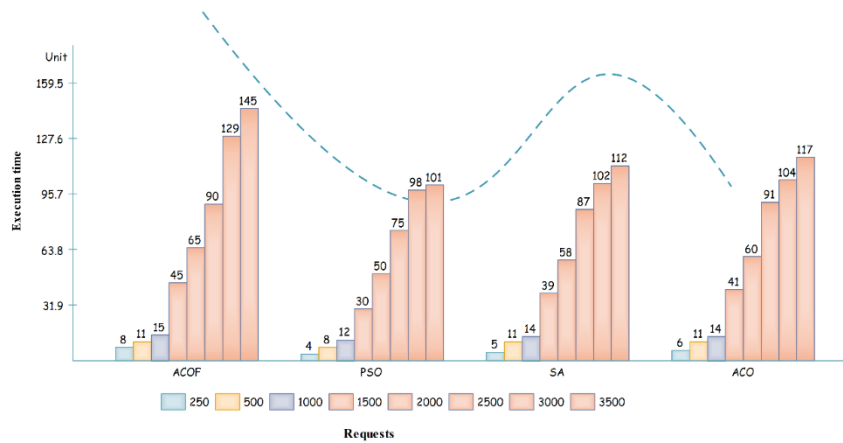
## 6. Experimental Results

This section presents the experimental results of resource provisioning for multi-tier clouds based on ACOF. Additionally, we conduct a comparison with other algorithms such as PSO, SA, and ACO. Table 1 presents the execution times for all algorithms.

**Table 1:** Results for Resources Provisioning Multi-tier Clouds based on ACOF, SA, And ACO Algorithms

Requests	Users	Datacenter	Execution time (sec)			
			ACOF	PSO	SA	ACO
250	3	1	8	4	5	6
500	5	1	11	8	11	11
1000	7	2	15	12	14	14
1500	10	3	45	30	39	41
2000	12	3	65	50	58	60
2500	13	6	90	75	87	91
3000	15	6	129	98	102	104
3500	16	8	145	101	112	117

As shown in table 1, performance diagram is illustrated in figure 7 separately for four resource provisioning meta-heuristic algorithms in dynamic state.



**Figure 7.** Evaluation of ACO, PSO, SA, and ACO Algorithms in Multi-Tier Clouds with Regard to Execution Time

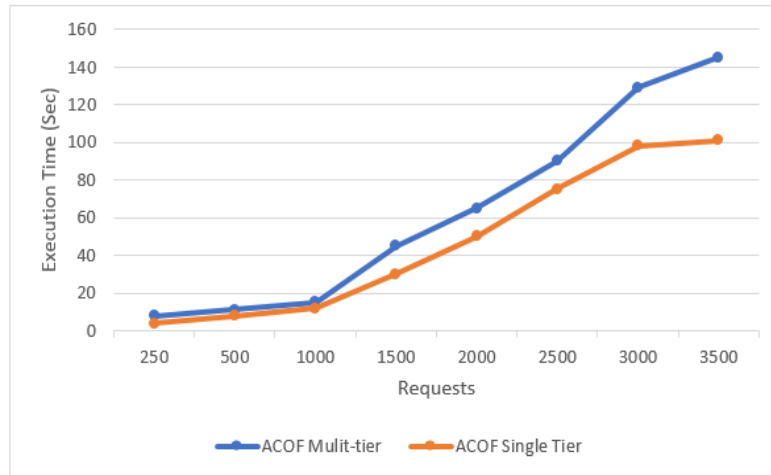
The figure illustrates that for fewer than 3500 tasks, the performance of the four algorithms is nearly identical, with the curves exhibiting no major differences, while congestion commenced at 3500 jobs. The algorithm that minimizes execution time can partially address this issue and allocate resources with reduced execution duration compared to alternative algorithms. After 5000 tasks, the program failed to process requests due to a bottleneck issue impeding its functionality.

Specifically, the PSO and SA algorithms exhibit a tendency to operate at a reduced speed when the quantity of tasks escalates. The ACOF and ACO algorithms initially exhibit sluggish performance; however, their efficiency enhances as the number of tasks increases, ultimately outperforming the SA and PSO algorithms in terms of time, bolstered by positive feedback.

The graphic clearly demonstrates that the ACOF algorithm outperforms the other algorithms. ACOF has a positive feedback mechanism, whereby the updating of available resources commences upon the selection of either a partial or a complete solution.

## 7. Performance

The rising utilization of Cloud computing services has rendered Quality of Service (QoS) a critical concern. The QoS qualities encompass response time, incurred cost, availability, and dependability. The aims of this study are to minimize overall execution time and optimize resource consumption, both of which directly affect costs. Furthermore, as the number of tasks increases, the disparities in performance become increasingly evident. Ultimately, when the task count reaches 3500, it is evident that the execution time of ACOF in Multi-tier Clouds is inferior to that in Single-tier Clouds, see figure 8.



**Figure 8.** Execution time of ACO Algorithm in Multi-tier and Single-tier

As shown in figure 8, the execution time of ACOF in Multi-tier Clouds is reduced by up to 13.2% compared to Single-tier Clouds, and the resource consumption and cost of the ACO algorithm in Multi-tier Clouds are lower than those in Single-tier Clouds are when the task number is 3500. Multi-tier Clouds enable the simultaneous execution of several requests through resource sharing.

The resource utilization (memory, CPU, HDD, and network) in the ACO algorithm for single-tier clouds remained constant throughout time. Initially, Multi-tier Clouds exhibited stability; however, they subsequently saw a decline due to the consumer's ability to share resources based on user requests.

## 8. Conclusion and Future works

The resource provisioning issue in a cloud-computing context has been framed as an optimization problem. The optimization solution to address the challenge must accommodate the varying responsiveness of diverse applications and manage dynamic fluctuations in resource requirements as workloads evolve over time. This study presents a dynamic resource provisioning meta-heuristic algorithm within a Multi-tier Cloud framework to address critical challenges in resource provisioning by minimizing total execution time and cost. The proposed method, termed ACOF, is a mix of Ant Colony Optimization (ACO) and the Firefly Algorithm (FA). This study conducted a comparison analysis of ACOF and four additional meta-heuristic algorithms. This study successfully optimizes resource provisioning by minimizing overall execution time and cost through the application of the ACOF algorithm in a multitier cloud environment. Consequently, resource provisioning reliant on ACOF algorithms can diminish execution time by up to 13.2% in comparison to the execution duration of ACOF algorithms in a single tier. The ACOF algorithm's advantage is in the utilization of a positive feedback mechanism, which initiates resource updating upon the selection of a partial or complete solution.

In the future, we should incorporate service reliability into the optimization formulas and improve the efficacy of the approaches.

## Acknowledgements

This article and the research behind it would not have been possible without the exceptional support of my University, University of Fallujah. In addition, the big effort, enthusiasm, knowledge and exacting attention to detail have been an inspiration by computer centre.

## References

- [1] H. Aiman and El-Maleh, "Binary particle swarm optimization (BPSO) based state assignment for area minimization of sequential circuits," *Applied Soft Computing*, vol. 13, pp. 4832–4840, 2013.
- [2] N. Ali Saoucha et al., "On Applying Firefly Algorithm for Cognitive Radio Networks," in *Proceedings of IEEE*, 2014; X. Cheng et al., "Cognitive Radio Spectrum Assignment Based on Artificial Bee Colony Algorithm," in *Proceedings of IEEE*, 2011.
- [3] M. H. Eawna, S. Hamdy, and E. -S. M. El-Horbaty, "New trends of resource provisioning in multi-tier Cloud computing," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, 2015, pp. 224-230.

- [4] J. Loganathan and S. Janakiraman, "An overview about major research problems in cooperative cognitive radio and its feasible solutions," *International Journal of Control Theory and Applications*, vol. 9, no. 19, pp. 9251–9265, 2016.
- [5] M. Freaj and A. Sleit, "Hybrid Approach for Resource Provisioning in Cloud Computing," *Journal of Information Systems and Communication Technology (JISCT)*, vol. 6, no. 1, pp. 546–561, Jul. 2016.
- [6] K. Sumalatha, M. N. Sinduri, and C. BhanuPrakash, "A Novel Method Of Directly Auditing Integrity On Encrypted Data," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, no. 1, pp. 1–5, Jan. 2016.
- [7] N. Grozev and R. Buyya, "Multi-Cloud Provisioning and Load Distribution for Three-Tier Applications," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 9, no. 3, Article 13, Oct. 2014.
- [8] M. H. Eawna, S. Hamdy, and E. -S. M. El-Horbaty, "Hybrid Algorithm for Resource Provisioning of Multi-tier Cloud Computing," *Procedia Computer Science*, vol. 65, pp. 682–690, 2015.
- [9] K. Chakraborty, A. Parashar, P. Bhambu, D. P. Tripathi, P. Patil, and G. K. Srivastav, "Multi-Level Cloud Datacenter Security Using Efficient Hybrid Algorithm," *Engineering Proceedings*, vol. 59, p. 50, 2023.
- [10] S. Kumar, U. Gupta, A. K. Singh, and A. K. Singh, "Artificial Intelligence: Revolutionizing Cyber Security in the Digital Era," *Journal of Computational Mechanics and Management*, vol. 2, pp. 31–42, 2023.
- [11] R. Manikandan et al., "Adaptive cloud orchestration resource selection using rough set theory," *Journal of Interdisciplinary Mathematics*, vol. 26, pp. 311–320, 2023.
- [12] A. Tiwari and R. Garg, "Adaptive Ontology-Based IoT Resource Provisioning in Computing Systems," *International Journal of Semantic Web and Information Systems (IJSWIS)*, vol. 18, pp. 1–18, 2022.
- [13] C. T. Dora Pravina, M. U. Buradkar, M. K. Jamal, A. Tiwari, U. Mamodiya, and D. Goyal, "A Sustainable and Secure Cloud resource provisioning system in Industrial Internet of Things (IIoT) based on Image Encryption," in *Proceedings of the 4th International Conference on Information Management & Machine Intelligence*, Jaipur, India, Dec. 2022, pp. 1–5.