Game Theory applies Ant Colony Optimization MetaHeuristic to balance the resource allocation in Cloud Computing environment.

Dang Dinh Minh, Dinh Quang Tung, Le Thanh Trang,
Tieu Duc Anh, Nguyen The Anh

April 2022

Abstract

Cloud computing is one of the most crucial modern conveniences, where resources are provided based on the user's demands. Cloud computing environment is a multi-object model for calculation that proposes a range of features for computing and storage based on user demand. Because of the increase of cloud users, the usage boosts highlight the issue of using and managing available resources. In this paper, we suggested a paradigm for resource allocation. based as a solution to achieve balance of the goals of stakeholders, which includes service providers and customers, based on game theory. Besides, we also used the Ant Colony Optimization meta-heuristic, which takes into account a number of cloud computing characteristics in order to efficiently allocate the appropriate computing resources for users in this environment [1]. This algorithm enables us to solve the identification of the problems associated with the task scheduling strategies in the cloud environment. Besides, it also helps us to determine the current configuration and information of physical servers via the pheromone line and to find the optimal or near-optimal resource allocation strategy between physical servers via the pheromone line and to find the optimal or near-optimal resource allocation strategy between physical servers. The experimental findings demonstrate the model's viability and efficacy.
Keywords: Cloud Computing; Resource Allocation; Game Theory; Unified Game-based Model; Ant Colony Optimization Meta-Heuristic

1 Introduction

1.1 Overview

Cloud computing is the Internet-based supply of information technology resources like as servers, memory, data, networks, applications, analysis, and intelligence on a pay-as-you-go basis [2]. It simply means that the resources are

totally controlled by a third-party provider rather than the end-user [3]. Cloud computing applies virtualization technology to divide large resources into many virtual resources. It will provide the underlying infrastructure and applications as a pre-made service and serve customers in a customized way to pay for what they use. Companies no longer concerned about human resources development, knowledge, or money for the upkeep of these resources thanks to the transition from on-premises software and hardware to a networked, distant resource [3]. The Infrastructure as a Service (IaaS) model provides users with infrastructures such as networks, servers, CPUs, memory, storage, and compute resources as virtual machines (VMs) with virtualization technology. Some examples are Amazon Web Services, VMware, Microsoft Azure Platform [3]. People discover that, while cloud computing provides ease, it also consumes a lot of energy when compared to traditional service models. Now, certain providers can adapt to dynamic changes in user demands in order to satisfy user expectations. However, in the real world, this will result in a significant waste of resources and an increase in wasteful energy usage. The application load request resources will alter in response to changes in user demand. As a result, reducing energy usage while maintaining service quality is an issue that must be addressed. Therefore, managing and using resources on cloud computing in an effective way is a significant problem. Because of the adaptability of Cloud Computing, service providers can quickly adapt and alter resource allocation based on user resource needs at any time to ensure the best quality of service and profits [4]. Most service providers strive to maximize their earnings at the lowest possible cost of investment, which leads to optimal resource utilization [4]. However, the maximization of natural resources, mainly the physical server, often leads to violations of service quality for customers. Furthermore, because the Internet is one of the cloud's foundations, network bottlenecks are unavoidable when enormous amounts of data are exchanged. In this approach, users are still responsible for resource management, but they typically have limited management tools and permission to address these issues [5]. On the other hand, failing to properly manage resources can lead to unwanted difficulties. One of the major consequences of improper resource management is the distrust of the clients. If - at any point - doubts are cast over the provider's ability to deliver adequate service, clients will consider making a negative review, moving to a different provider, or even changing to a different model entirely. Creating a cascading effect that leads to a decline in both user base and profits, resulting in bankruptcy and a major loss of invaluable data [6]. Therefore, service providers will try and look for different actions and strategies. Here are some of the issues that need to be resolved:

Conflict	Description
The conflict in resources	Number of Processing Units (Cores) Architecture (64 bit or 32 bit) Main memory (RAM) System Bus Speed (Gb/second) Storage (Secondary Memory)
The conflict between several decision-makers	Between consumer and service provider Between physical servers Between virtual machines From different goals of consumers

Table 1: The conflicts of allocating process of cloud computing resources [3]

Game theory is used to investigate scenarios involving two or more people in which the result of one of their actions is dependent not only on that person's unique behavior but also with the actions of the other participants in the game. Based on the information accessible to them, the theoretical conclusion of the game is described as strategic combinations that are more likely to meet the goal of the participants. Balanced tactics (Equilibrium Strategy) of the player without any motor players who vary their action plans are known as strategic combinations [4]. The balance of a game describes the strategies that reasonable players are predicted to choose when they interact with each other. However, individuals with limited information about the intended actions (strategies) of others, must make educated guesses about what the others will do. This type of thinking is known as strategic thinking, and we will use it to help us understand the core of the problems, predict the possible outcomes and suggest solutions to mitigate or completely fix the issues. Our paper follows two main methods to solve the resource allocation issue as follows:

(i) Prove the feasibility and efficiency of the model by using Ant Colony Optimization to find the Nash equilibrium of the problem.

(ii) Propose a new solution that includes players, their strategies, their payoff function, and Nash equilibrium based on Game Theory, especially the Unified Game-based Model.

In section 2, we will introduce some publications related to the resource allocation problem and the algorithms mentioned. For the rest of the paper, we divided the content into five parts to point out the description of the resource allocation problem as follows. In section 3.1, the resource allocation problem was given. Then, we built the conflict of resource allocation in section 3.2. Next, in section 3.3, an example solution solving resource allocation was proposed. In section 3.4, we introduced our model which is based on Game Theory, especially the Unified Game-based Model. Finally, in section 3.5, we published a solution using the Ant Colony Optimization algorithm.

2 Literature review

One of the primary challenges in the cloud computing environment is resource allocation. For this reason, there are various studies related to this topic. The solution to the resource allocation problem is frequently based on the individual features of each situation, from which techniques such as the comprehensive algorithm, determinism algorithm, or metaheuristic algorithm can be used. For example, Fei Teng et al. used the exhaustive algorithm [5], while Morton applied the deterministic algorithm [6]. In the meantime, it is necessary to have the ability of extension and the ability to meet high consumer conditions in cloud computing, which is an environment with distributed data. Ghanbari et al. suggested a work scheduling method based on priority for usage in cloud computing. Multiple qualities and multi-criteria judgments are taken into account [7]. Polverini et al. proposed the optimum energy cost and queuing delay limitations. After then, adhere to the server's maximum input temperature constraints. They've shown that with future data, it can reduce arbitrary costs close to the optimal offline algorithm [8]. Alejandra et al. advocated using meta-heuristic and metaheuristic optimization to minimize execution costs via scheduling. [9]. Keshk et al. recommended modifying ant colony optimization for load balancing. This strategy shortens the duration of a work. This approach takes neither the availability of resources nor the weight of jobs into account [10]. Accordingly, it is possible to approach the problem of scheduling virtual machines on the Cloud in a metaheuristic direction like Ant Colony Optimization is feasible because this algorithm can give nearly-optimal results in an adequate time. Over the most recent couple of years, numerous asset assignment issues were addressed by game theory in distributed computing. Non-cooperative games were investigated for the virtual machine balance and placement challenges by Ye and Chen [11]. They attend to the Nash equilibrium and concern not much about another way for the best allocation plan. Furthermore, Hassan et al solved the challenge of distributed resource allocation in cloud computing using both the non-cooperative and cooperative game models [12]. The results show that the cooperative game strongly motivates the service provider to contribute resources. However, their resources only come from a single source, while the allocation must be concentrated on many resources. Another method is max-min fairness, which maximizes the minimum number of resources that each user gets. From the above method, Waldspurger has improved and added weights to accommodate policies that depend on various factors such as deadlines, priorities, and reservations [13]. However, these two studies only focused on the equity of a single type of resources, while ignoring other types. In terms of the multiple types of resources, Ghodsi et al pointed out a dominant resource fairness approach (DRF) that computes the dominant share of each user to address the issue [14]. However, in their study, there is still an unsatisfactory point because they do not consider the waste of resources. Cloud computing environments have a large number of physically heterogeneous servers, so efficient resource allocation is also an interesting topic to consider. To increase server resource usage, Steinder et al. developed a

data center management system and applied resource allocation to a heterogeneous mix of workloads [15]. Besides, a novel scheme Dynamic Optimization Problems (DOPs) was described in order to enhance resource use and execution efficiency by Di and Wang [16]. Deshi Ye and Jianhai Chen [11] have built a non-cooperative game model between virtual machines, which considers two factors: (1) Issue with server task scheduling, and (2) Problem with virtual machine location. In the above model, there is a migration of the virtual machine to another physical machine if more appropriate resources are available. However, there is a big obstacle that virtual machines and physical machines are not identical in capacity, configuration, etc. According to Zhen Xiao et al, the allocation of resources in the data center is depending on the demands of the application, and the calculation to optimise the number of systems in use may be generated using technology virtualization [17]. Be that as it may, this strategy just accomplishes neighborhood streamlining. However, complete streamlining isn't accomplished. This article just arrangements with the issue of portion upon demand. Pandit et al introduced an effective asset portion calculation utilizing Simulated Annealing [18]. Regardless, Simulated Annealing is a technique utilized for single-objective advancement, so it isn't reasonable for multi-objective issues. Chonho et al. [19] presented the Nudge calculation that acknowledges usages to change the positions and asset circulations to the ecological circumstances in a cloud. Besides that, a heuristic approach has been devised by various authors to handle job scheduling and resource allocation difficulties. Radojevic et al. developed a centralized task scheduling decision model for usage in cloud settings, automating the scheduling process and reducing the involvement of human administrators. However, this model falls short in assessing node capabilities and configuration specifics, and the entire system lacks backup, resulting in a single point failure. Furthermore, Goswami et al. concentrate on work scheduling while taking into account various restrictions. A better approach to work on the speed of undertakings in the Map/Reduce type has been shown by Chochan et al [20]. The following table is some highlight publications and their factors:

Authors	Publication	Factors
Ye and Chen [7]	Non-cooperative games on multidimensional resource allocation	Nash equilibrium Non-cooperative game
Hassan et al. [8]	Game-based distributed resource allocation in horizontal dynamic cloud federation platform	Non-cooperative game dynamic cloud federation
Qinghong Sang [10]	A Resource Trusted Model Based on Game Theory in Cloud Computing	Trusted model The rewarding mechanism
Waldspurger [12]	Lottery and Stride Scheduling: Flexible Proportional-Share Resource Management	Proportional-share mechanisms
Ghodsi et al [13]	Dominant resource fairness: fair allocation of multiple resource types	Dominant resource fairness (DRF) Max-min fairnes
M. Steinder [15]	Server virtualization in autonomic management of heterogeneous workloads	Heterogeneous workloads
Di and Wang [16]	Dynamic optimization of multiattribute resource allocation in self-organizing clouds	Heuristic algorithms Dynamic Optimization Problems (DOPs)
Zhen Xiao et al. [17]	Dynamic resource allocation using virtual machines for cloud computing environment	Prediction algorithms

Table 2: Some highlight publications and their factors

After a detailed analysis, we found that the above articles were all complete and detailed. However, in our opinion, the above articles still have some problems that need to be raised and resolved as follows: [5], [6], [17] have not completely solved the problem of conflicts between many types of resources and different requirements from users in a cloud computing environment; [11], [12] do not achieve maximum efficiency with inappropriate models such as non-cooperative model, trusted model; [13], [18] only focus on a single-objective like single type of resource or single physical server, so it is not suitable for the requirement of multi-objective environments such as cloud computing and Game Theory.

Due to the drawbacks of the aforementioned research, as well as a number of difficulties with scheduling in the cloud computing infrastructure, including load balancing, financial, and calculation time concerns [20], we have learned

from their experience and decided to apply the model due to the Ant Colony Optimization algorithm to provide a new solution that can solve the conflicting problems and optimize the model's performance. People might look for some actions dealing with balancing issues since Ant Colony Optimization (ACO) is a type of intelligent biological algorithm utilized to identify the ideal answer by modeling the eating behavior of ants. It accomplishes this by distributing the workload throughout the cloud architecture and reducing computational overhead. Makespan in cloud computing refers to the moment when the final job completed and left the system. ACO controls the amount of clouds, memory use, and virtual machine usage [20]. Because of their natural and chemical properties, each moving ant always leaves a "phenomenon trail" in its path, and they often follow the path of stronger odors. These "pheromone trails" are chemicals that evaporate. Initially, the amount of odor in the two branches was approximately the same. After a certain time, the short branches will have a stronger smell than the longer ones. The amount of odor is approximately the same but the distribution of the longer branch is less dense than that of the odor in the shorter ones. Because it will evaporate faster than the amount of smell on the branch for longer at the same time. An ant's intellect is low, yet it achieves optimal population efficiency by exchanging knowledge among groups, acting as a kind of hive mind. The ant colony algorithm is a positive feedback algorithm with moral fortitude and adaptability. Furthermore, it may be used with other algorithms to swiftly find more appropriate answers [21]. As a result, the Ant Colony Optimization method outperforms other algorithms in tackling combinatorial optimization problems. The advantage of using ACO in this game is determining the reinforcement information via the pheromone line. We can determine the information that the pheromone line represents. The pheromone trail is the ability of a consumer's preferred physical server to assign virtual machines on demand based on the server's current configuration and heuristic information [22].

3 Problem Description

3.1 Resource allocation problem

Resource allocation faces major problems, some of which include cost effectiveness, reaction time, redistribution, computing performance, and job scheduling. At the same time, cloud computing service users desire to do tasks at the lowest feasible cost [23]. The resources in cloud service providers' data centers are growing more diverse, consisting of different generations of infrastructure hardware. Cloud service providers must embrace these technological changes in order to keep their services current, but this exacerbates the problem of heterogeneity [4]. As a result, it's critical for cloud service providers and consumers to take advantage of the constantly growing heterogeneity in order to achieve their goals: the most efficient use of resources and cost control. Cloud service providers, on the other hand, face a number of challenges when allocating resources among

users' tasks based on their application usage patterns. Customers want cloud service providers to anticipate their application requirements, and they also expect the task to be accomplished on schedule [4]. Besides, Adnan Abid et al. [24] also mentioned that in the case that a work takes longer than expected, the service provider must plan for the availability of resources. As a result, a strategy for dealing with interruptions while also transferring the operation to an available resource is necessary. Furthermore, resource users' estimations of resource requirements to accomplish a project before the expected time may result in over-provisioning of resources. The allocation of resources by resource providers may result in resource under-provisioning [25]. Effective resource allocation approaches are necessary to overcome this problem. While consumers require effective services to ensure the timely delivery of their application data. Because of rising energy prices and the need to reduce greenhouse gas emissions and overall energy consumption, communications, and storage, efficient resource allocation is one of the key issues in cloud computing [24].

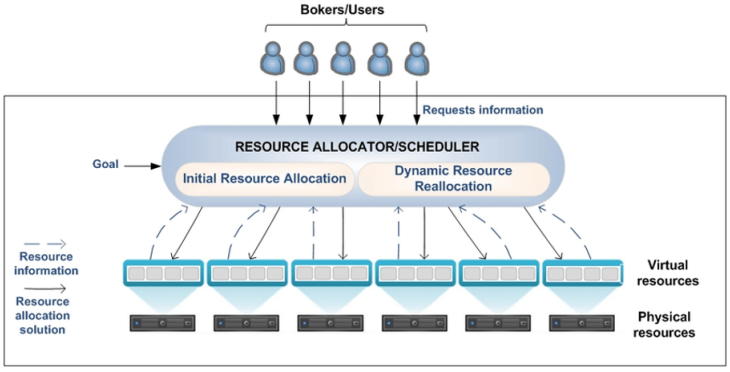

Figure 1: Resource Allocation in Cloud Computing [26]

In order to provide efficient services, the qualities and properties of both the cloud service provider and the cloud service consumer should be considered, with the objective of allocating adequate resources to an appropriate task with the target of the task being finished on time and the cloud provider making the most profit, as shown in the preceding discussion.

3.2 Examples of problem

The fairness in resource allocation is applied from the concept of DRF (Dominant Resource Fairness) researched in a multi-resource environment by Ghodsi et al [17]. Furthermore, Xu et al [30] have proposed a fair resource allocation model that still ensures efficient use of resources. In other words, they minimize the level of resource waste to ensure the fairness of the resource allocation decision, which is calculated using the formula:

$$V(A) = (\sum_i \sum_j (\left| \frac{x_{ij}}{c_j} - \tau * d_{ij} \right|)^{\alpha-1})^{\frac{1}{\alpha}} \ (\alpha \epsilon R) \ (1)$$

In which:

- V: Set of players - physical servers have resources available to respond to user's requests.
- A: Set of user resource requirements
- a: Resources are allocated on request by the server
- F: Set of player utility functions
- x_{ij}: is the number of resources j allocated to the request i
- $\tau = \frac{1}{max \sum_i d_{ij}}$ is the dominant share which is the largest proportion of any given resource allocated to the user.
- $d_{ij} = \frac{Y_{ij}}{max Y_{ij}}$ is a standardized requirement.

The usage of the resource cost of the physical server n is calculated according to the following formula:

$$w(n) = \sqrt{\sum_j \left(\frac{u_j^{(n)}}{u^{(n)}} - 1\right)^2} \quad (2)$$

In which:

- $u_j^{(n)}$ is resource j used by nth physical server.
- $u^{(n)}$ is the average of the resources used by the physical servers.

The utility function is built on a fair and efficient allocation of resources. Each player will choose a strategy to maximize their own gains. The utility function of a resource allocation decision is calculated as follows:

$$F^n(A) = \beta * v(A) + (1 - \beta) * w(n) \quad (3)$$

where $\beta \in [0, 1]$ is a weight indicating the level of concern for fairness or efficiency in resource use.

3.3 Characteristics of players, strategies

In this study, we apply the game model to solve the problem of allocating cloud computing resources. In this model, players are virtual machines which have a few characteristics including storage, CPU, memory, network bandwidth, resource, task, time, cost. The storage is available storage space, especially a warehouse's allotted space. CPU is electrical circuitry that executes a computer program's instructions. Memory or RAM provides a temporary storage location for apps to store and retrieve data. Network bandwidth is the maximum quantity of data is transferred over the Internet connection for a certain length of time. Resource is a part of capacity of physical resources that behave like physical resources but are provided on demand rather than in advance of

need. Task indicates system actions not completed immediately. Time suitable for computer systems is closely combined, perform tasks from relevant languages and target tasks into new platforms when running. Cost is usage charge.

Besides, there are some strategies such as Service Level Agreement (SLA), hardware resource requirement, policy, execution time, virtual machines, utility. SLA is the agreement made between the end user và the cloud service provider prior to the start of communication. Hardware resource requirement is application-specific hardware resource usage. Policy is certain conditions and parameters must be considered in order to achieve optimal resource utilization and load balancing results, and policies can be used to achieve this. Execution time is the amount of time necessary to complete the task in the cloud. Virtual machines is computer systems that are established by running software on one physical computer to mimic the operation of another physical computer. Utility is improved performance, faster reaction time, and cheaper cost.

Property	Characteristics
Player: Virtual machine	- *Storage* - *CPU* - *Memory (RAM)* - *Network bandwidth* - *Resource* - *Task* - *Time* - *Cost*
Strategy	- Service Level Agreement (SLA) - Hardware resource requirement - Policy

Table 3: Summarize properties characteristics

3.4 Dataset

The length of time it takes the server to respond to a request is determined by network capacity, the number of clients, the amount of requests, and the average think time. The following formula determines the response time at CPU's peak load (in seconds):

$$T_{response} = n/r - T_{think} \quad (4)$$

Where:

- $T_{response}$ is the response time.
- T_{think} is the average think time in seconds per request.
- n is the number of users.

- r is the number of requests the server receives per second

	Storage	Internal Network Bandwidth	CPU	GPU	RAM	Users	Requests	Think time (t_{think})	Response time ($t_{response}$)
VM 1	8GiB	Up to 3 Gbps	Intel Xeon Scalable Processor (Ice Lake) 3rd Generation	NVIDIA A100	8GB	1000	100	3	7
VM 2	16GiB	Up to 5 Gbps	Intel Xeon E7 (Broadwell E7)	NVIDIA T4	16GB	1000	150	2.5	4.17
VM 3	32GiB	Up to 8 Gbps	Intel Xeon E5 v4 (Broadewell E5)	NVIDIA V100	32GB	1000	200	2	3

Figure 2: Problem without game theory applied

	Storage	Internal Network Bandwidth	CPU	GPU	RAM	Users	Requests	Think time (t_{think})	Response time ($t_{response}$)
VM 1	8GiB	Up to 3 Gbps	Intel Xeon Scalable Processor (Ice Lake) 3rd Generation	NVIDIA A101	8GB	1000	260	3	0.85
VM 2	16GiB	Up to 5 Gbps	Intel Xeon E7 (Broadwell E7)	NVIDIA T5	16GB	1000	310	2.5	0.73
VM 3	32GiB	Up to 8 Gbps	Intel Xeon E5 v4 (Broadewell E5)	NVIDIA V101	32GB	1000	320	2	1.13

Figure 3: Problem with game theory applied

With game theory applied, the resource allocation is well-divided, which increases the amount of requests from users that a server can receive, reducing the response time from the server. All three virtual machines listed above benefit from the strategy. A server now can take more requests without upgrading its hardware components.

4 Model

4.1 Introduction to Unified Game-based model

The number of physical servers: nS

The Unified Game-Based Model is defined as follows:

$$G = <(P_0, P), (S_0, S_i), (u_0, u_i), R^c> \quad (5)$$

In which:

- P_0 : Physical server.
- $S_0 = S_{01}, \ldots S_{0j}, \ldots S_{0k0}$ is a set of Physical server strategies.
- u_0 : from S_0 to R is the payoff function of the Physical server.

- $P_i = P_1, \ldots, P_m$: A set of consumers.
- $S_1 = S_{i1}, \ldots S_{ij}, \ldots S_{imi}$ is a set of Consumer's strategies with $i(1 <= i <= m)$. The number of resources in which user i-th participates is denoted by m_i. The solution to this problem can be thought of as a user registration strategy for cloud storage resources.
- u_1 : from S_{ij} to R is a benefit of the consumer.
- R_c: is the vector space representation of the problem

4.2 Developed a mathematical model for this problem with explanation

The Physical server strategy $S_0 = S_{01}, \ldots S_{0j}, \ldots S_{0k0}$. Each S_{0j} should be satisfied with some of the characteristics in Figure 3.

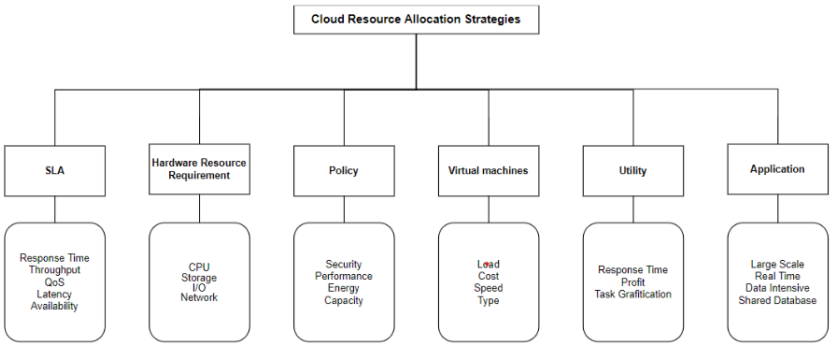

Figure 4: Resource allocation strategies in Cloud Computing Environment [29]

Based on the resource allocation strategies above, each resource allocation decision of the physical server 0 can be represented as a matrix:

$$S_{oj} = \begin{vmatrix} D_{11} & D_{12} & ... & D_{1j} \\ D_{21} & D_{22} & ... & D_{2j} \\ ... & ... & ... & ... \\ D_{s1} & D_{s2} & ... & D_{sj} \end{vmatrix}$$

In which, each row of the matrix corresponds to the resource allocated according to the requests. Dsj is the resource allocated to the request of the physical server. In order for virtual machines to be distributed fairly across physical servers, use the formula:

$$H_0 = \frac{U_0}{i_0} \quad (6)$$

- H_0: Resource utilization efficiency of the physical server 0

- U_0: Resources used in the physical server 0
- i_0: Total resources of the physical server

$\longrightarrow$ System load balance:

$$: L = \frac{\sum_{i=1}^{n}\left(H_0 - \overline{H}\right)}{n} \ (\overline{H}\text{: } Average\ value\ of\ efficiency)$$

In order for service providers to achieve high profits: exploit the maximum service capacity of the physical server, avoiding the waste of resources of the physical machine.

Wasted resources of the physical server 0: $w_0 = \frac{A_0}{i_0}$ (8)
A_0: Available resources of the physical server 0
Total system exploitation level: $W = \sum_{i=1}^{n} w_n$ (9)

- The $S_1 = S_{11}, \dots S_{1j}, \dots S_{1m}$. S_{1j} represented by the vector P having the dimension $S_{1j} = S_{1j1}, \dots S_{1jk}, \dots S_{1jh}$, h is provider strategy number of items in category j, and S_{1jk} is the information for the item k of category j.

The Payoff of:

- Physical server: $F_{ps} = (k_1 * L_1 + (1 - k_1) * W_1)^{t_1} + (k_2 * L_2 + (1 - k_2) * W_2)^{t_1+t_2} + (k_n * L_n + (1 - k_n) * W_n)^{t_1+t_2+\dots+t_n}$
- Consumer: $F_c = k_1^{t_1} + k_2^{t_1+t_2} + ... + k_n^{t_1+t_2+\dots+t_n}$
  $\longrightarrow$The payoff of both: F = C*F$_{ps}$ + $D * F_c$

In which,

- k is the number of resources that the physical server 0 intends to allocate to the consumer.
- C,D: tuning constants
- t: Time needed to complete

4.3 Explanation of Nash Equilibria formula and the application of Nash Equilibria in this topic

Nash Equilibrium is a game theory concept that defines the ideal solution in a non-cooperative game in which no single player can maximize profits if the other players' strategies are likewise fixed [29]. When player i chooses the j^{th} strategy, if it is the optimal strategy as indicated by S_{ij}^*, the optimal strategy of the other players is as indicated by S_{-ih}^* then the strategy's Nash equilibrium will adhere to the condition, as follows:

$$U_i(S_{ij}^*, S_{-ih}^*) \geq U_i(S_{ij}, S_{-ih}^*) \ (10)$$

4.4 Proved the optimal solution from algorithm is a NE by using Nikaido Isoda function

H. Nikaido and K. Isoda (1955) proposed the Nikaido-Isoda function, which determines the Nash equilibrium in conflict, to generalize the Nash equilibrium issue in non-cooperative games. The function is written in the following form in the resource allocation problem:

$$f(x^*, x) = \sum_{i=1}^{n} (f_i(x) - f_i(x[y_i])) \quad (11)$$

where, vector x[yi] is vector that formed by transferring from vector x to xi by yi. The strategy set of player i is as indicated by Ki R. At this moment, the strategy of the game will be K = K1x. . . .xKn and Nash equilibrium in the game x* K happens if and only if:

$$f_i(x^*) = max f_i(x * [y_i]), y_i \epsilon K_{i'} i \quad (12)$$

When $f_i(x^*)$ satisfies formula 1, the Nash equilibrium is determined using the Nikaido-Isoda function. In general, the Nash equilibrium must also match other criteria, including as the problem data and to determine the Nash equilibrium using the Nikaido-Isoda function, the problem will get a multi-objective optimum form and will require multi-objective evolutionary algorithms (MOEA) to solve.

5 Algorithm with NE

5.1 Why can this algorithm be used in solving the problem?

The ant's natural behavior serves as the basis for the ant colony optimization. Ants forage near their nests on a daily basis. The ant accumulates a chemical called a pheromone as it travels in search of food. It makes it possible for other ants to return home. It also allows other ants to see their previous path to food, allowing them to follow in their footsteps. Because the pathway with the highest pheromone concentration is the shortest and leads to the most pathway migrations, it will deposit more pheromones on it. This is how ants converse in order to determine the quickest path to their destination. The ants take longer to travel from source to destination when the pheromone levels drop. Over time, the pheromone trace fades or diminishes. Therefore, to build a solution, probabilistic construction employs both pheromones and problem-specific empirical data. Each component can only be chosen if it isn't already in use, and those components must be chosen from the current set of components [27].

The benefits of this technology include: reducing peak energy consumption of computing resources, saving energy during downtime, regularly utilizing environmentally friendly energy sources, mitigating the negative effects of computing resources, and reducing resource waste [27].

5.2 How can this algorithm find Nash Equilibria?

Equilibrium can be unstable in a multi-agent system context. Furthermore, finding the Pareto-efficiency of Nash equilibrium is difficult. The majority of the algorithm is built on the algorithm meta-heuristic to solve this problem. The plan allocates VMs to discover the best solution using ant colonies methods [30]. The optimal plan will be chosen from a viable plan based on Nash equilibrium criteria. When no player can get a reward that is greater than the estimated near ideal, it signifies that all players have chosen their approximated Pareto optimal tactics. If F_j^{itr} presents the payoff of player j in iterator itr of the ant colony optimization algorithms,$F_j^{itr} - F_j^{itr-1}$ presents the payoff improvement of player j. The termination criterion is the total of the square deviations of all players' payoffs that are smaller than a certain number e, i.e.:

$$\sum_{i=1}^{n}(F_j^{itr} - F_j^{itr-1})^2 \leq \varepsilon \quad (13)$$

5.3 Explanation of some formulas / parameters of algorithm can be customized to solve the problem

Tasks are substantially resourced in such a way that the overall execution time of the tasks must be decreased. The allocation of resources occurs in three stages:

- Understand user needs
- Initialize parameters
- Allocate resources

The tasks and resources of all processes are located in heterogeneous locations. In the cloud system, the resource allocation is done flexibly by the application according to the requirements. Various planning approaches are used for resource provisioning and planning. By allocating time intervals to each job, multiple allocation of resources to tasks in a quite way that the overall execution time and cost of the activities are minimized [30].

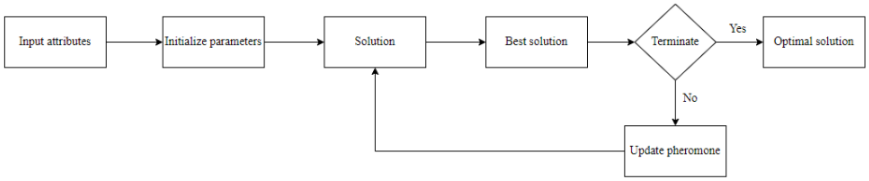

Figure 5: The process using ACO based resource allocation [27]

The process of resource allocation using ACO is as follows:

- The input properties of the number of processes, tasks, and resources are noted down.

- Ants were used for the process and traces of pheromones used by the ants were recorded. Two pheromone pathways are used in this algorithm. The first path is used to select the shortest path, i.e. pheromone trail intensity, and the other represents the desired resource selection ability.
- During this construction phase, the ant allocates resources to the task based on the probabilistic selection rule expressed as pheromones and heuristic information.
- Probabilistic selection rules are used. if there is any conflict during scheduling. The current best solution (i_{best}) and global best solution (g_{best}) are recorded.
- After a specified number of iterations, check if the best solution is reached. In case best achieves a poor solution, the solution is reconstructed to obtain an optimal solution.

In this way, ACO finds the optimal solution and allocates resources efficiently [27].

5.4 Diagram and Psuedo code of customized algorithm to find Nash Equilibria

Input: List of Cloudlet (Tasks) and List of VMs
Output: the best solution for tasks alloction on VMs

Steps:

1. Initialize:

   Set Current iteration t=1.
   Set Current optimal solution=null.
   Set an initial value $\tau_{ij}(t) = c$ for each path between tasks and VMs.

2. Place the m ants on the starting VMs randomly.
3. For k :=1 to m do

   Place the starting VM of the k-th ant in $tabu_k$.
   Do ants trip while all ants don't end their trips
   Every ant chooses the VM for the next task according to formula (1).
   Insert the selected VM to $tabu_k$.
   End Do

4. For k :=1 to m do

   Compute the length Lk of the tour described by the k-th ant according to formula (4).
   Update the current optimal solution with the best founded solution.

5. For every edge (i,j), apply the local pheromone according to formula (5).
6. Apply global pheromone update according to formula (7).
7. Increment Current iteration t by one.

8. If (Current iteration $t < t_max$)

Empty all tabu lists.
Goto step 2
Else
Print current optimal solution.
End If
Stop

6 Conclusion

In this article, we have shown how to use game theory for the cloud computing resource allocation problem, as well as provide information about the conflicts and strategies applied. We used the Unified Game-Based Model that was presented to solve the existing situation. This work proposes the Ant Colony resource allocation technique for the cloud environment's large-scale, sharing, and dynamic features, so that computer resources may be distributed properly. Based on the simulation results, it is clear that this algorithm is capable of searching for and distributing resources in a cloud computing environment. The implemented algorithm takes care of efficiently allocating resources to tasks. The major focus of this study is energy conservation; the ACO algorithm minimizes the amount of energy required by each resource, improving system performance. Efficient use of time is also made using this algorithm. The results show that the execution time is reduced. However, due to some limitations, we just implemented the ant colony algorithm. In future studies, we will compare and evaluate other metaheuristic algorithms such as genetic algorithms, particle swarm optimization algorithms, etc will be researched and implemented.

REFERENCES

[1]: Microsoft, " What is cloud computing?" https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/

[2]: What Is IaaS, PaaS, And SaaS? Examples And Definitions: A Cloud Report

[3]: Bui Thanh Khiet et al, "MO HINH CAP PHAT TAI NGUYEN IEN TOAN AM MAY CONG BANG DUA TREN LY THUYET TRO CHOI", (2017)

[4]Fei Teng. "Resource allocation and scheduling models for cloud computing", (2011)

[5]Morton, T., and Pentico "Heuristic scheduling systems: With applications to production systems and project management", (1993),

[6] Shweta Varshney, Sarvpal Singh, "A Survey on Resource Scheduling Algorithms in Cloud Computing", (2018)

[7] Shamsollah Ghanbari, Mohamed Othman, "A Priority Based Job Scheduling Algorithm in Cloud Computing", (2012)

[8]: Marco Polverini; Antonio Cianfrani; Shaolei Ren; Athanasios V. Vasilakos, “Thermal-Aware Scheduling of Batch Jobs in Geographically Distributed Data Centers”, (2014)

[9]: Maria Alejandra Rodriguez; Rajkumar Buyya, “Deadline Based Resource Provisioningand Scheduling Algorithm for Scientific Workflows on Clouds”, (2014)

[10]: Keshk, Arabi E.; El-Sisi, Ashraf B.; Tawfeek, Medhat A., “Cloud Task Scheduling for Load Balancing based on Intelligent Strategy” [11] D. Ye and J. Chen, “Non-cooperative games on multidimensional resource allocation”, (2013)

[12] M. Hassan, B. Song, and E. N. Huh, “Game-based distributed resource allocation in horizontal dynamic cloud federation platform”, (2011)

[13] Rajkumar Buyya and Manzur Murshed, “GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing”, 2002.

[14] Qinghong Shang, “A Resource Trusted Model Based on Game Theory in Cloud Computing” (2020)

[15] Hadoop, “Scheduling in Hadoop”, 2012,

[16] C. A. Waldspurger, “Lottery and Stride Scheduling: Flexible Proportional-Share Resource Management”, (1995)

[17]. A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, “Dominant resource fairness: fair allocation of multiple resource types”, (2011)

[18] D. C. Parkes, A. D. Procaccia, and N. Shah, “Beyond dominant resource fairness: extensions, limitations, and indivisibilities”, (2012)

[19] M. Steinder, I. Whalley, D. Carrera, I. Gaweda, and D. Chess, “Server virtualization in autonomic management of heterogeneous workloads”, (2007)

[20] S. Di and C. L. Wang, “Dynamic optimization of multiattribute resource allocation in self-organizing clouds”, (2013)

[21]. Xiao, Z., Song, W., and Chen, Q.: ”Dynamic resource allocation using virtual machines for cloud computing environment”, (2013)

[22] Diptangshu Pandit, Samiran Chattopadhyay, Matangini Chattopadhyay and Nabendu Chaki, “Resource allocation in cloud using simulated annealing”, (2014)

[23] Chonho et el, “An evolutionary game theoretic approach to adaptive and stable application deployment in clouds”, 2010.

[24] Chohan N, Castillo C, Spreitzer M, Steiner M, Tantawi A, Krintz C. ”See spot run: Using spot instances for MapReduce workflows”, 2010.

[25] Jayden Kiprotich, “Resource Allocation Algorithms and Strategies in Cloud Environments”, 2021

[26]: Xin Guo, “Ant Colony Optimization Computing Resource Allocation Algorithm Based on Cloud Computing Environment”, (2016)

[27]: Zheng-Tao Wu, “Application of Ant Colony Optimization in Cloud Computing Load Balancing”, (Oct 2017)

[28]: Adnan Abid et al., “Challenges and Issues of Resource Allocation Techniques in Cloud Computing” (2020)

[29]: Rabi Prasad Padhy, “Service Support Aware Resource Allocation Policy for Enterprise Cloud-based Systems” (2013)

[30]: Harshitha H.D., Beena B.M., “Ant Colony Optimization for Efficient Resource Allocation in Cloud Computing” (2017)

[31]: Vinothina, V., R. Sridaran, and Padmavathi Ganapathi. ”A survey on resource allocation strategies in cloud computing.” International Journal of Advanced Computer Science Applications 1 (2012): 97-104.

[32]: Jayden Kiprotich. “Resource allocation algorithm and strategies in Cloud Environment” (2021)

[33]: Ghribi, C. (2014). Energy efficient resource allocation in cloud computing environments (Doctoral dissertation, Institut National des Télécommunications).

[34]: Corporate Finance Institute. (2022). Nash Equilibrium.

[35]: Trinh Bao Ngoc, “AP DUNG LY THUYET TRO CHOI VA CAN BANG NASH XAY DUNG PHUONG PHAP MO HINH HOA XUNG DOT TRONG QUAN LY DU AN DAU TU CONG NGHE THONG TIN VA THU NGHIEM TRONG MOT SO BAI TOAN DIEN HINH”, Hanoi 2020

[36]: Nikaido H., Isoda K. (1955), ”Note on noncooperative convex games”, Pacific Journal of Mathematics, 5, pp. 807-815.