Application of Game Theory to allocate resources in Cloud Computing by using NSGA-II algorithm

Le Hong Hai, Le Huy Hoang, Do Thanh Dat, Tran Thi Ngoc Anh, Pham Tuan Anh, Tran Diem Quynh
Hanoi University, Faculty of Information Technology
Hanoi, Viet Nam
July 9, 2022,

Abstract

Cloud computing is one of the thriving and present-day innovations today because of its utilities. A Cloud computing environment is a multi-object model for calculation that proposes a range of features for computing and storage based on user demand. To accomplish this goal, many cloud specialist cooperatives face the test of working with clients with every one of the essential assets as indicated by their necessities. Thus, the focus of this paper is to address resource allocation for optimizing energy consumption and cost in the cloud. With the application of Game Theory, we could make certain all resources will be distributed fairly and efficiently when the model reaches Nash equilibrium. In the research, the model that we pick is the "Unified Game-based Model". Likewise, we proposed another component known as the NSGA-II algorithm. This has demonstrated the effectiveness and efficiency of multi-objective evolutionary algorithms in addressing multi-objective problems. In this study, the resource allocation problem is solved using NSGA-II. Empowered to track down an ideal asset allotment by utilizing game-theoretic ideas. The experimental findings demonstrate the viability and efficiency of the aforementioned model.

Keywords*: cloud computing, game theory, resource allocation, Unified Game-based Model, NSGA-II algorithm.*

1 Introduction

1.1 Overview

Cloud computing and cloud-based applications are progressively gaining popularity. This is a kind of virtualization that permits on-request network admittance to process assets. This model saves money because instead of owning computer resources, consumers lease them from a service provider when they are needed. Furthermore, cloud computing will provide mobile users access to computer resources, a concept known as mobile cloud computing. This is critical since mobile devices are increasingly becoming consumers' primary computing platforms, despite their limited processing power and battery life. Cloud computing permits clients to share PC assets powerfully. The cloud is a huge assortment of helpfully open and valuable virtual assets. With everything being viewed as a service, it is a service-delivery model that offers users a selection of flexible and efficient services [1], [2]. The fundamental tenet of cloud computing is that user data is stored online rather than locally [4]. Because cloud computing offers a variety of resources and services, it has become more and more popular among cloud users. A focused association is made possible by a number of notable distributed computing frameworks, such as those offered by Amazon, Microsoft, IBM, Google, and Hewlett-Packard [5], [3]. These frameworks enable engineers to spread applications among PCs. The majority of cloud service providers construct systems with. Users' services are allotted to the cloud service provider based on their ongoing consumption requirements, and the cloud service provider has the flexibility to alter resource allocation to ensure service quality and profitability. As a result, providing and maintaining cloud resources efficiently is a big problem. Good resource allocation in cloud computing is essential for effective resource utilization, a large number of physical servers, and charging users to utilize them.

Cloud service providers may change resource allocation based on user resource needs over time to guarantee service quality and profit. Most service providers try to maximize income in the lowest possible investment costs, helping to make optimal use of resources. However, the maximum use of natural resources - here is the physical server often leads to a violation of the quality of service for the customer. Therefore, service providers will have different actions and strategies. Here are some issues that need to be addressed:

Conflict	Description
The conflict in resources	Storage (Gb) CPU (number of cores) RAM (main memory) Bandwidth (Gb/s) The speed of the network (Mb/s) Construction (32-bit/ 64-bit)
The conflict between stakeholders	Between user and cloud provider Between VMs (virtual machines) Between PMs (physical machines) Between usage among users Between the goal of cloud providers

Figure 1: Some conflicts in the process of allocating cloud computing resources [38]

Moreover, this study will analyze and criticize some recent resource allocation optimization works in terms of solutions and ways to handle the current difficulties. Then, to give an optimized method for resource allocation in cloud computing infrastructures that offers better trade-offs than existing solutions, various studies that have been done so far utilizing CloudSim as a cloud simulator tool will be presented. It will also demonstrate some of the findings and consequences, as well as the likely reasons and impacts on performance, memory, and CPU, as well as the overall allocation process. Then, to create an adaptive resource allocation solution that minimizes power consumption, and optimizes the use of the cloud model's benefits.

With the aid of this article, the following questions will be addressed: How do you allocate resources efficiently and precisely? What is the most effective method for resolving cloud optimizing issues? Resource allocation is anticipated to yield higher performance, less power consumption, be more time and money efficient, and result in less resource waste. Finally, research and outcomes, as well as specific case studies, have been done to acquire a better grasp of the current situation through the use of various allocation techniques.

2 Literature Review

A variety of domains in cloud computing need to be optimized and developed, so many scientists and researchers have been considering and improving to meet the demands of providers and consumers. The Hybrid Particle Swarm Optimization approach is used by Gomathi and Karthikey [8] to assign work in a distributed context (HybPSO). It suits user needs and boosts load balancing while increasing productivity. The object is to give load adjusting by diminishing position fulfillment time among processors. Using this strategy, each task is dedicated to a single processor. The Nudge approach, developed by Chonho [12], enables cloud applications to adapt their locations and resource distribution to shifting environmental conditions. In order to reduce the time and expense of load adjusting by utilizing assets to achieve the Pareto ideal front, Xue [13] presents a non-dominated Pareto sorting genetic algorithm-based multi-objective method (NSGA-II). However, the algorithm's strategy for further developing effectiveness and execution has not been successful, as the solutions are trapped in local optimum. In a different publication, Chandrashekhar S. Pawar and Rajnikant B. Wagh [14] developed a system that took into account the execution of Preemptable tasks as well as numerous SLA criteria, including the amount of memory and CPU time needed, network bandwidth, and other factors. Their findings showed that our strategy maximizes resource usage when resource contention is high. In [15], Cheng uses a general heuristic methodology to create the best hierarchical resource allocation solution for workflows. This model's main objective is to make sure that the duties and obligations assigned to the service are coordinated. In order to ensure that tasks are executed correctly and that priorities are upheld, it is important to deliver service in accordance with operational needs. A multi-objective work scheduling approach for distributed computing systems based on fuzzy systems and NSGA-II algorithms was proposed by Salimi [16]. The authors' study on load balancing in distributed systems is driven by their desire to shorten implementation times and costs while increasing resource productivity. They use fuzzy systems, the indirect approach, and the creation of the third objective function to address this problem. However, they have struggled to manage three objectives in their job. A unique load balancing technique employing HoneyBee and Ant Colony Behavior Algorithms in Cloud Environments is proposed by Mousavi & Fazekas in [38]. The suggested approach aims to balance the workload of the virtual machines while attempting to cut down on job completion times and speed up cloud infrastructure response times. For the load balance and virtual machine placement challenge, Ye and Chen examine noncooperative games [39]. They are more interested in the existence of Nash equilibrium than the best allocation strategy's solution. In the federated cloud, Hassan [10] presents a method for the distributed resource allocation problem that considers both cooperative and noncooperative games. They demonstrate that in the cooperative allocation game, suppliers are more inclined to provide resources. Our study, on the other hand, takes the allocation problem into account in a multi-resource setting, whereas their work only analyzes the problem in a single-resource context. The table below lists some major publications and their contributing elements:

Authors	Publications	Methods
Gomathi and Karthikey [8]	Task Scheduling Algorithm Based on Hybrid Particle Swarm Optimization in Cloud Computing Environment	Hybrid Particle Swarm
Chonho et al. [12]	An evolutionary game-theoretic approach to adaptive and stable application deployment in clouds	Nudge method
S Xue [13]	An Improved Algorithm Based on NSGA-II for Cloud PDTs Scheduling, Journal of Software	NSGA-II & Pareto Sorting Genetic algorithm
Chandrashekhar S. Pawar [14]	Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue	Modified Waiting Queue
B. f Cheng [15]	Hierarchical Cloud Service Workflow Scheduling Optimization Schema using Heuristic Genetic Algorithm	Heuristic Genetic Algorithm

Figure 2: Some highlight publications and their factors

After all, literature reviews reveal that traditional methods can be conclusive and precise, but they frequently have various drawbacks, such as low performance, time cost, user authentication protocols, resource status, and loading time and balance when employing cloud resources. As a result, we carried out this research to solve the remaining weaknesses using the theoric game strategy and the Genetic algorithm. Essentially, game theory is used to examine scripts involving two or further persons in which the outgrowth of one of their acts is told not just by that existent's conduct, but also by the conduct of other players in the game. Because of the data given to them, the hypothetical finish of the game is portrayed as an essential blend probably going to meet the members' objectives. Assuming there are no players with a motivation to change their ways of behaving, key blends are considered to address the player's balance techniques. Because of these variables, we accept GT is the best answer for addressing distributed computing's issues. At the point when there is an appropriate virtual machine asset portion methodology, it will help with the more productive and stable usage of assets. Resource allocation issues can also be solved using methods like the exhaustive algorithm, and deterministic algorithm, as well as the simulated annealing approach [9]. Exhaustive algorithms usually outperform deterministic algorithms in experiments. Deterministic algorithms, however, are ineffective in dispersed data environments, making them unsuitable for resource allocation issues in a scaled context [10]. Cloud computing, on the other hand, is a distributed information environment that necessitates adaptability as well as a high capacity to meet client requirements. Thus, we developed a novel technique known as the NSGA-II algorithm. Additionally, cloud service providers will have the ability to efficiently and timely distribute resources, ensuring that cloud clients are happy with the services they receive.

3 Problem Description

3.1 Description of problem in 1-2 paragraphs (7-10 lines), citation for terminologies used in the description of problem

One of the most contentious issues in cloud computing is resource distribution. The distribution of resources from resource producers to resource consumers is the main objective of resource management in order to maximize their utilization [23]. Technologies for virtualization help multiplex resources, enabling more efficient resource use. There are four different categories of resources: computing, networking, storage, and power [23].

The goal of resource allocation, a part of resource management, is to distribute available resources as effectively as possible. Due to two main considerations, namely the requirement to meet resource demands in a range of environments and the continually changing demands of cloud users, the challenge of allocation of resources administration has become more significant [24]. To allocate resources, a variety of task scheduling approaches are used. A group of techniques known as scheduling is used to manage the sequence in which computer systems carry out actions [25]. One of the numerous scheduling algorithms that have been created is task scheduling. It can be difficult to schedule a group of tasks submitted by different users on a set of computer resources in a cloud environment in order to speed up job completion [26]. Therefore, the following criteria should be avoided in order to maximize resource allocation: [26]

- Resource conflict happens when two apps attempt to access the same resource simultaneously.
- When resources are in short supply, resource scarcity is present.
- Resource fragmentation happens when resources are isolated. Although there will be enough resources, they cannot be allocated to the necessary application.
- When an application is given more resources than it needs, this is known as over-provisioning of resources.

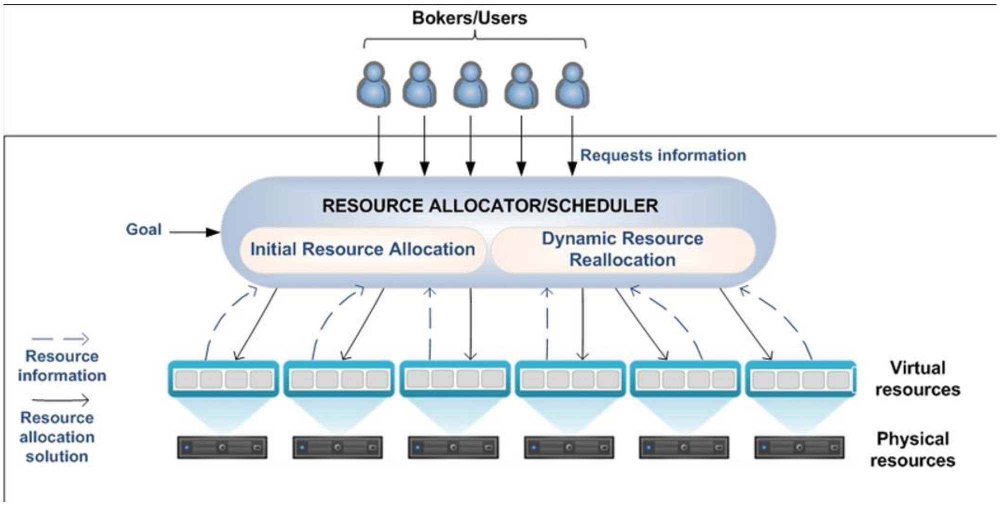

Figure 3: Resource Allocation in Cloud Computing [12]

The main challenge in cloud computing is making sure that resources are efficiently provided to cloud applications through the Internet to encourage the effective and cost-effective use of cloud resources [27] to reduce costs and increase throughput. Effective cloud resource allocator methods or rules are necessary to ensure that cloud users are satisfied with the services provided by being able to allocate properly and just in time. [24]

3.2 Had at least 1-2 examples of problems, including citations for these problems

We consider the physical server as the player in a game. The player's strategy is to decide the resource allocation of each physical server based on the resource requirement set by the user. It is assumed that while allocating the resource, it has n demands: *D1…Dn* (n>=1). For each of *Di* (*i*: from 1 to n), it has a variety of methods for handling *Hij*. [29]

In which, *Hij* is the j-th method for requesting *Di* (*i:* 1 → *n*; *j*: 1 → *m*; *m* and *n* >= 1) Requests related to each other, when handling *Di* by the *Hij* method, will probably create a conflict with the other *Dm* of *Hmp* (conflict between resources). And below is a figure that describes the model of the problem.

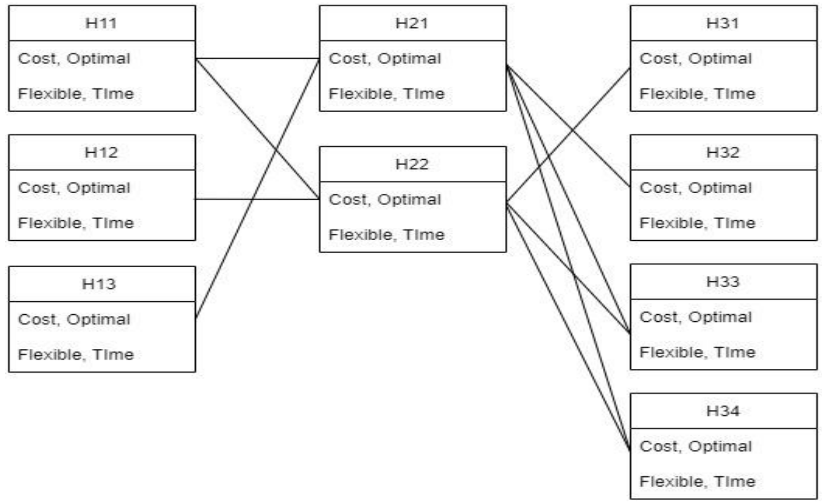

Figure 4. The model of the problem in resource allocation (cloud computing) [29]

Four fundamental aspects distinguish each demand technique of resource allocation:

- *Cost*: The amount of money (in VND) necessary to meet the demands.
- *Optimal*: reduce work pressure (run many applications, heavy programs), optimize resource allocation method, compared to other methods.
- *Flexible*: based on the “Rapid Elasticity” of cloud computing: automatically expand and shrink the system depending on the users' needs. [30]. In other words, it means the balance of each resource allocation method of handling other ones.
- *Time*: the period that the technique was used to meet the demand (hours).

In conclusion, to handle conflicts of resource allocation is to discover the path that goes through every single one of the demands Di with the method of handling. No conflict here.

3.3 Defined all characteristics of players, strategies, which will be the input of algorithm, highlight them

In this paper, it is considered that a Virtual machine is a player. There are some characteristics of virtual machines listed below:

CPU time: how much time the CPU has spent specifically processing data for a particular program or process
Memory (RAM): gives applications a place to store and access data on a short-term basis
Energy Consumption: the amount of energy or power used
Cost: operating cost

The strategies of the virtual machine are listed below:

Service Level Agreement (SLA): before communication begins, this is an agreement made between the cloud service provider and the end-user.

Hardware Resource Requirement: utilizing hardware resources in accordance with the demands of the program.

Execution Time: is the amount of time required to finish the task in the cloud.

Policy: certain circumstances and characteristics must be allowed to acomplish the best outcomes for resource usage and load balancing, and for this, the rules can be used.

Utility: optimized for greater performance, quicker reaction, and cheaper cost

3.4 Used at least 1 table/figure to summarize/ analyze these characteristics

Properties	Characteristics
Player: Virtual machine	CPU time Memory (RAM) Energy Consumption Cost
Strategies	Service Level Agreement (SLA) Hardware Resource Requirement Execution Time Policy Virtual machines Utility

Figure 5: Properties, characteristics, and strategies

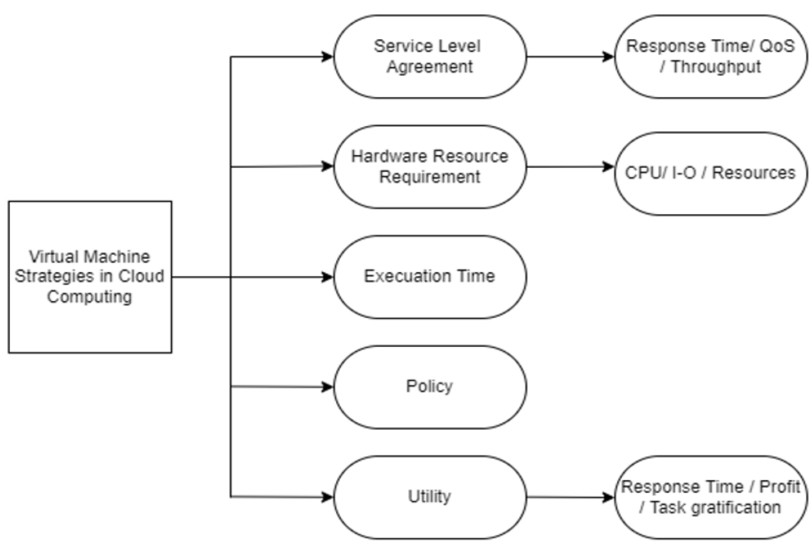

Figure 6: Virtual machines strategies in cloud computing [24]

3.5 Used the above examples or a new one to find Nash equilibria with a sampled data set, used formula/variable to explain the process

According to author Beloglazov [43], he proposed a model to solve the problem for energy-aware related to resource allocation by the formula as follow:

$$P_{(u)} = x \cdot P_{max} + (1 - x) \cdot P_{max} \cdot u \quad (1)$$

Where,

P_{max}: the maximum consumption power used by a virtual machine when it is wholly utilized.

x: the percentage of energy used by an idle server (in other words 70 percent).

u: the CPU utilization

In the paper, we approach the problem as a multi-objective issue with two potentially incompatible goals: cutting down on overall web service costs and using physical equipment energy [41].

The model is modeled as 2 vital resources: CPU time C = {C_1, C_i, …, C_t}, memory consumption D = {D_1, D_i, …, D_t}. C_i quantity of CPU time, as well as D_i number of memory, are consumed for each request. The coming rate as frequency is described as E = {E_1, E_i, …, E_t}. The model of physical machines in cloud data center are designated as CPU time PC = {PC_1, PC_j, …, PC_p}, memory PD = {PD_1, PD_j, …, PD_p}. The CPU volume of a PM (physical machine) is indicated as PC_j. And, the sizableness of memory is indicated as PD_j. A physical machine can be virtualized or divided into a number of virtual machines. Each VM has its own CPU time VC = {VC_1, VC_k, …, VC_z}, as well as memory VD = {VD_1, VD_k, …, VD_z}.

It is defined that X_k^i is service allocation's decision variable. And, it is a binary value (0/1) indicating whether or not a service i is assigned to virtual machine k. It is also defined that Y_j^k is VM allocation's decision variable. And, it is a binary value (0/1) showing whether or not a VM k is deployed on PM j.

In the problem, it is considered that PM is homogenous which means PM has the same value of memory and CPU time. It is indicated that U = {U_1, U_k, …, U_z} shows the CPU utilization of a VM. It can be calculated by the following formula:

$$U_x = \begin{cases} \frac{\sum_{i=1}^{t} E_i \cdot C_i \cdot X_k^i}{VC_k}, & If \quad \sum_{i=1}^{t} E_i \cdot C_i < 1 \\ 1 & , otherwise \end{cases} \quad [41]\ (2)$$

It is defined that O = {O_1, O_k, …, O_z} showing the cost of VM type. In order to satisfy the performance requirement, service providers often define Service Level Agreements (SLAs) to ensure service quality [41]. If a VM reaches its utilization limitation, it means that the services have been allocated exceedingly. Therefore, all services in that VM suffer from performance degradation. The overall energy usage and the entire cost of operating virtual machines are defined as two optimization methods:

$$\text{Energy} = \sum_{j=1}^{p} \left(x \cdot V_{max} + (1-x) \cdot V_{max} \,.\, \sum_{k=1}^{z} U_k \cdot Y_j^k \right) \quad [41]\ (3)$$

$$\text{Cost} = \sum_{j=1}^{P} \sum_{k=1}^{z} O_k \cdot Y_j^k \quad [41]\ (4)$$

Here, there are two datasets to compare before and after applying game theory to the problem.

	CPU time (s)	RAM (Gb)	Energy Consumptio n (Watt)	Cost (Vnd)
VM 1	1	8	12.23	48.92
VM 2	2	16	24.34	97.36
VM 3	3	32	37.56	150.24

Figure 7: Problem example without game theory applied

	CPU time (s)	RAM (Gb)	Energy Consumptio n (Watt)	Cost (Vnd)
VM 1	1	8	8.56	34.24
VM 2	2	16	17.03	68.15
VM 3	3	32	26.29	105.16

Figure 8: Problem example with game theory applied

It can be seen that there are 3 available VMs. 3 VMs are virtualized from PM with 3 different values of CPU time and RAM. Before applying game theory, the energy consumption of VM1, VM2, and VM3 are 12.23, 24.24, and 37.56 correspondingly. According to formula (1), after the game theory was applied, the energy consumption of VM1, VM2, and VM3 are 8.56, 17.03, and 26.29 correspondingly. It means that the energy consumption decreases by 30% compared to before the game theory was applied. Also, before the game theory is applied, the cost of VM1, VM2, and VM3 are 48.92, 97.36, and 150.24 correspondingly. According to formula (2), after the game theory applied, the cost of VM1, VM2, and VM3 are 34.24, 65.15, and 105.16 correspondingly. It means that the cost decreased by 30% compared to before the game theory was applied.

So, it is effective when applying game theory to optimize the energy consumption and cost of the virtual machine.

4 Game Model

4.1 Introduction to Unified Game-based model (if the topic isn't related to Matching theory)

According to author Trinh Bao Ngoc [37], The latest model (Unified Game Based Model) is suggested based on game theory and Nash equilibrium to model conflict in project management, by owner/non-owner. In the game related to investment, this player will be the investor. On the other hand, in a non-investment game, this player's role would be the referee. At the same time, the influence of the factors in each player's strategy will be shown through a feature vector. Thus, by combining three factors: special players, normal players, and all-player conflict, the model based on the Uniform Game are described as follows:

$$G = <\{P_0, P\}, \{S_0, S_i\}, \{u_0, u_i\}, R^c>[37]$$

Where

G: perform the game

P_0: the special player who can be the stakeholder or the referee to ensure the interests of all players.

$S_0 = \{s_{01}, \dots, s_{0j}, \dots, s_{0Ho}\}$: special player's strategy; Ho is the number of the strategies of the special player.

u_0: the payoff function derived from the special player's real-number tactics.

$P = \{p_1, \dots, p_i, \dots, p_K\}$: normal player, and K is the total of the players.

$S_i = \{s_{i1}, \dots, s_{ij}, \dots, s_{iHi}\}$: normal players' strategies; Hi is the total of the ordinary players' strategies.

u_i: a payoff function from the normal players' strategies in real numbers.

R^c: a vector representing the conflicts between the strategies of K players in problems.

4.2 Developed a mathematical model for this problem with explanation

Here, this model is especially optimal when applied to the problem of resource balance - the problem without an investor, which plays a decisive role in the activities of the organization. And it also creates a premise for the application of the algorithm to find the Nash equilibrium.

The problem of resource balance:

$$G = <\{P_0, P\}, \{S_0, S_i\}, \{u_0, u_i\}, R^c> [37]$$

Where,

P_o: cloud provider as the special player who represents the interests of the project investor. (Explanation: Because the energy consumption is reduced, the cost of maintaining the server will decrease, and the supplier will benefit)

$S_0 = \{s_{01}, \dots, s_{0j}, \dots, s_{0Ho}\}$: the set of cloud provider strategies, including Ho is the number of unallocated resources

u_0: $S_0 \rightarrow R$ is the payoff function of the investor that references a particular player's strategy to real numbers

K: Number of groups/departments/departments with resource requirements

$P = \{p_1, \dots, p_i, \dots, p_K\}$ is a set of virtual machines.

$S_i = \{s_{i1}, \dots, s_{ij}, \dots, s_{iHi}\}$: is a set of virtual machine strategies with i ($1 <= i <= K$). And H_i is the number of resources that player i participates in.

u_i: $S_i \rightarrow R$: is the payoff function of player i, referencing the strategy of player i to a real number.

R^c: is a vector space representing C the conflicts of the problem, and a non-empty vector $\vec{v} \in R^c$ represents the conflict between T parts of the project, having resource requirements with the same characteristics of skills, salary, or having the same request time to join $(1 \leq T \leq K)$.

The strategy of cloud provider:

Set of cloud provider's strategies: $S_0 = \{s_{01}, \ldots, s_{0j}, \ldots, s_{0Ho}\}$.
A special player's s_{0j} strategy (representing the cloud provider) has the following component: cost.

The strategy of virtual machines:

For the strategies of player *i*, which are parts of the project with resource requirements, the description of the player i strategies is of the form: $S_i = \{s_{i1}, \ldots, s_{ij}, \ldots, s_{iHi}\}$: is the set of strategies of the player i ($1 <= i <= K$) and *Hi* is the number of resources that the player participates in. s_{ij} has the following components: CPU time, RAM, energy consumption.

Payoff function of the virtual machine:

The payoff function of player *i* which is the *i-th* division out of total N parts in cloud computing is determined as follows:

$$u_i = \sum_{j=1}^{P} \left(x \cdot V_{max} + (1 - x) \cdot V_{max} \,. \sum_{k=1}^{z} U_k \cdot Y_j^k\right)$$

In there,

P: consumption power

V_{max}: VM reaches its utiization limitation

x: the percentage of energy used by an idle server (in other words 70 percent).

U_k: CPU utilization of a VM. Calculated by formula (2)

Y_j^k: VM allocation's decision variable

The payoff function of the cloud provider:

The payoff function of the added virtual player which represents the interests of the project investor is determined as follows:

$$u_0 = \sum_{j=1}^{P} \sum_{k=1}^{z} O_k \cdot Y_j^k$$

In there,

P: consumption power

O_k: the cost of VM type

Y_j^k: VM allocation's decision variable

The fitness function of the problem solution:

The fitness function value of the problem is calculated based on the player's two payoff values, the higher the fitness value, the better the solution.

$$u_{all} = u_0 + \sum u_i$$

Constraint:

+) Hard restriction: A virtual machine can only be allocated to a physical server if and only if it has sufficient capacity for all of its resources. [41]

$$\sum_{k=1}^{z} VD_k \cdot Y_j^k \leq PD_j$$

$$\sum_{k=1}^{z} VD_k \cdot Y_j^k \leq PC_j$$

+) Soft restriction: Although service can be given to a virtual machine even if that machine does not have sufficient capacity for all of its resources, the quality of the services that are allocated will deteriorate. [41]

$$\sum_{k=1}^{z} VC_k \cdot Y_j^k \leq PC_j$$

4.3 Explained the Nash Equilibria formula, and its application in the field of 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 [37]. When player i chooses the j^{th} strategy, if it is the optimal strategy as indicated by s_{ij}^*, the effective strategy of other players is as indicated by S_{-ih}^* then the strategy's Nash equilibrium will add here to the condition, as follows:

$$U_i\left(s_{ij}^*, s_{-ih}^*\right) \geq U_i\left(s_{ij}, s_{-ih}^*\right) \text{ [37]}$$

4.4 Proved the optimal solution from the 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:

$$f(x_1^* x) = \sum_{i=1}^{n} \left(f_i(x) - f_i(x[y_i])\right) \text{ (1) [37]}$$

where vector $x[y_i]$ is a vector that is formed by transferring from vector x to x_i by y_i. Signed at $K_i \subset R$ is player i-th's strategies. At this moment, the strategy of the game will be $K \coloneqq K_1 \times \ldots \times K_n$ and Nash equilibrium in the game $x^* \in K$ happens if and only if:

$$f_i(x^*) = \max_{y_i \epsilon K_i} f_i(x^*[y_i]) \text{ , } \forall y_i \in K_i \text{ , } \forall i \text{ [37]}$$

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

About the meaning, the Nikaido Isoda function is widely used in applications to solve general equilibrium problems, as well as Nash equilibrium, which shows that if the multi-objective optimization algorithm converges, it will find Nash equilibrium. In this model, the function has solved (proved the correctness, proved the solution is correct) the problem of optimal load balancing, optimizing energy consumption and operating cost in the cloud computing infrastructure.

5 Algorithm

5.1 Explained why this algorithm can be used in solving the problem

NSGA-II has demonstrated the effectiveness and efficiency of multi-objective evolutionary algorithms in addressing multi-objective problems. A popular MOEA which has been successfully applied to numerous optimization issues in the real world is NSGA-II. In this study, the resource allocation problem is also solved using NSGA-II. We first suggest a representation, and then we provide an approach using novel genetic operators and the NSGA-II.

5.2. Explained how this algorithm can find Nash equilibria

The theory of cooperative and non-cooperative games is based on Nash Equilibrium. It's a collection of activities built on a balanced approach with traits that prevent any one player from deviating from the prescribed method and obtaining greater rewards. The crucial factor in any situation is how to select the strategy that will produce the best results. Below is the table to describe how NSGA-II finds Nash equilibria in this problem.

Factors	Description
Initialization	This process aims to produce a population that is diversified. Finding the best VM type to run each service in the first stage depends on the resource needs of the service. In the second stage, a stronger type is randomly created based on the appropriate VM type. Additional information is given in pseudocode in the 5.4.
Mutation	Individuals should be able to explore the entire search space that is feasible, according to the mutation operator's design tenet. As a result, a good mutation operator has the capacity for exploration as well as the capacity to maintain a person within the boundaries of possibility. Additional information is given in pseudocode in the 5.4
Violation control method	Since the chromosomal representation immediately removes the hard constraint, a changed violation ranking is suggested to address the soft constraint. The quantity of services allocated in the downgraded virtual machines is how we specify a violation number. That is, if a VM has too many services allocated, the performance of all of the services will deteriorate. The number of violations is used for the selection process, and those with fewer violations are almost always favored.
Selection	Our strategy takes advantage of the constrained-domination principle along with a binary tournament selection. The following is a definition of a constrained-domination principle. If any of the following conditions hold true, the solution I is deemed constraint-dominant over solution J: 1) Solution I is possible whereas the solution is not, 2) Both solutions are impractical, but I have fewer total violations, and 3) Both solutions are practical, but solution I predominates over solution J. Always choose someone with no violations or fewer violations. This approach has been demonstrated to be successful in the original NSGA-II study [49].
Fitness function	The kind of VMs that are assigned to web services affects the cost of fitness (formula 4). Formula 3 illustrates energy fitness. According to the ratio of the CPU capacities of the VMs and PMs, the VM utilization (from formula 2) is first translated into the PM utilization.

Figure 9: How NSGA-II finds Nash equilibria in this problem [39]

5.3 Explained some formulas/parameters of the algorithm can be customized in order to solve the problem

The parameter of the algorithm can be described in the table below:

Parameter	Description	Note
Crossover probability	0.8	(N stands for the number of design parameters)
Mutation probability	1/N	
Crossover distribution index	20	
Mutation distribution index	20	
Number of iterations	1000	
Population size	100	

Figure 10: Parameter for NSGA-II algorithm

5.4 Included both diagram and Psuedo code of customized algorithm in order to find Nash equilibria

Algorithm 1 Initialization
Inputs:
Virtual machine CPU Time V C and memory V D,
Service CPU Time C and memory D
Consolidating element c
Outputs: A service allocation population

for Each service t **do**
    Find the best VM Type for it.
    Create a VM type virtualmachineType at random that is equal to or better than its most appropriate type.
    if VMs with the specified virtualmachineType already exist **then**
        Create a random number u
        if u < consolidating element **then**
            select at random one current VM with virtualmachineType to allocate
        else
            create a new VM using virtualmachineType
        end if
    else
        Make a new VM using the most appropriate VM type.
    end if
end for

Algorithm 2 Mutation
Inputs:
An individual VM CPU Time V C and memory V D,
Service CPU Time C and memory D
consolidating element c
Outputs: A individual with mutation

```
for Each service do
    Make a number u at random
    if u < mutation rate then
        determine which VM Type is best for this service.
        Create the number k at random.
        if k < consolidating element then
            determine the used VMs' utilization
            Give each virtual machine (VM) a fitness rating of 1 / utilization and create a roulette wheel using fitness values.
            Create a number p at random, then choose the VM using that number.
            Distribute the service
        else
            start a new virtual machine with the best VM Type
            Place the new VM in a spot that is chosen at random.
        end if
    end if
end for
```

Algorithm 3 NSGA-II algorithm
Inputs:
Virtual machine CPU Time V C and memory V D,
Physical Machine CPU Time P C and memory P D,
Service CPU Time C and memory D
Consolidating element c
Outputs: A set of solutions that is non-dominated

```
Set up a population P
while Termination Condition is not satisfied do
    for Each individual do
        Analyze the fitness metrics
        Determine the violation
    end for
    non-Dominated Sorting of P
    determine crowding distance
    while child number is fewer than the population size do
        Selection
        Mutation
        add the child in a new population U
    end while
    Merge U and P {for elitism}
    Determine the combined U and P
    For the combined population, non-dominated sorting and crowding distance
    Include the highest-ranking popSize users in the following generation.
end while
```

6 Conclusion

Cloud computing is a rapidly evolving technology, and numerous studies have been undertaken to address the issues it presents. The cloud faces a multitude of barriers and one of these is the resource allocation techniques for all users. Therefore, this paper is based on the NSGA-II algorithm to address this problem. We identified that including the NSGA-II algorithm in resource distribution in cloud computing was very effective in supporting the cost-effective and efficient use of cloud resources for the purpose of optimizing energy consumption and cost. Furthermore, we also use game theory in this research. In detail, we proposed an instance of the game model in resource allocation (cloud computing), building the conflict model of cloud computing resource allocation. And finally, we proposed a model based on the Unified Game-Based Model to handle the conflict (in the resource allocation form). After this research, we learned a lot of things related to cloud computing, game theory, and algorithm (NSGA-II). This is a basis for us to do good research in the future.

LIST OF FIGURES

REFERENCES

[1]. Swapnil M Parikh, Dr. Narendra M Patel, Dr. Harshad Kumar, B Prajapati, *"Resource Management in Cloud computing: classification and taxonomy."*, 2017.

[2]. Pandaba Pradhan , Prafulla ku. Behera, BNB Ray, *"Modified Round Robin Algorithm for Resource Allocation in cloud computing"*, ELSEVIER Procedia Computer Science, 2016.

[3]. Pandaba Pradhan , S. Pani , *“A New Approach for Resource Allocation in Cloud Computing”* , International Journal of Computer Science And Technology (IJCST), 2014.

[4]. Navpreet Kaur Walia, Navdeep Kaur, *“Resource Allocation Techniques in cloud computing: A Review”*, 4th International Conference on Advancement in Engineering and Technology(ICAET),2016.

[5]. B. Jennings, R. Stadler, *“Resource Management in the cloud: Survey and Research challenges”*, Journal of Network and Systems Management, 2015.

[6]. Morton, T., and Pentico, D. W.: „Heuristic scheduling systems: with applications to production systems and project management‟ (John Wiley & Sons, 1993. 1993)

[7]. Van Laarhoven, P. J., Aarts, E. H., and Lenstra, J. K.: *"Job shop scheduling by simulated annealing"*, Operations research, 1992

[8]. B. Gomathi and K. Karthikeyan: *Task Scheduling Algorithm Based on Hybrid Particle Swarm Optimization in Cloud Computing Environment*, Journal of Theoretical and Applied Information Technology, 2013

[9]. Colorni, A., Dorigo, M., Maniezzo, V., and Trubian, M.: *"Ant system for job-shop scheduling"*, Belgian Journal of Operations Research, Statistics and Computer Science, 1994

[10]. Tsai, C. W., and Rodrigues, J. J.: *"Metaheuristic scheduling for cloud: A survey"*, IEEE Systems Journal, 2014

[11]. Eugen Feller, Lavanya Ramakrishnan, Christine Morin, “*Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study, Journal of Parallel and Distributed Computing*”, Elsevier, 79-80, pp.80-89, (2015).

[12]. Chonho Lee, Junichi Suzuki, Athanasios Vasilakos, Yuji Yamamoto, and Katsuya Oba., *“An evolutionary game-theoretic approach to adaptive and stable application deployment in clouds”*, In Proceeding of the 2nd workshop on bio-inspired algorithms for distributed systems, ACM, New York, 2010.

[13]. Pandaba Pradhan , S. Pani , *“A New Approach for Resource Allocation in Cloud Computing”* , International Journal of Computer Science And Technology (IJCST), 2014.

[13]. S. Xue: *An Improved Algorithm Based on NSGA-II for Cloud PDTs Scheduling, Journal of Software*, 2014

[14] Chandrashekhar S. Pawar and Rajnikant B. Wagh “Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue”, International Conference on Intelligent Systems and Signal Processing (ISSP), pp.311-316, 2013

[15]. B. f Cheng, “*Hierarchical Cloud Service Workflow Scheduling Optimization Schema using Heuristic Genetic Algorithm”*, Telecommunications, 2012.

[16] R. Salimi, H. Motameni, and H. Omranpour: Task Scheduling Using NSGA II with Fuzzy Adaptive Operators for Computational Grids, Journal of Parallel and Distributed Computing, Vol. 74, 2014

[16]. D Sumathi, *“Max-Min Task Scheduling Algorithm in Cloud Computing”*, Telecommunications, January 2020.

[17]. Upendra Bhoi, Purvi N.Ramanuj, *“Enhanced Max-min Task Scheduling Algorithm in Cloud Computing”*, International Journal of Application or Innovation in Engineering Management, April 2013.

[18]. Li, J., Qiu, M., Ming, Z., Quan, G., Qin, X., and Gu, Z, *‘Online optimization for scheduling preemptable tasks on IaaS cloud systems’,* Journal of Parallel and Distributed Computing, 72, (5), pp. 666-677, 2012.

[19]. Rahman, M., Li, X., and Palit, *'Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment',* in Editor (Ed.)^(Eds.): 'Book Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment' (IEEE, edn.), pp. 966-974, 2011.

[20]. Saovapakhiran, B., Michailidis, G., and Devetsikiotis, *'Aggregated-DAG scheduling for job flow maximization in heterogeneous cloud computing',* in Editor (Ed.)^(Eds.): 'Book Aggregated-DAG scheduling for job flow maximization in heterogeneous cloud computing' (IEEE, edn.), pp. 1-6, 2011

[21]. Pandaba Pradhan, Prafulla Ku. Behera, B N B Ray, *"Enhanced Max-Min Algorithm For Resource Allocation In Cloud Computing",* International Journal of Advanced Science and Technology, 2020.

[22]. Pandaba Pradhan , Prafulla ku. Behera , BNB Ray , *"Modified Round Robin Algorithm for Resource Allocation in cloud computing"* , ELSEVIER Procedia Computer Science, 2016.

[23]. B. Jennings, R. Stadler , *"Resource Management in cloud: Survey and Research challenges"* , Journal of Network and Systems Management , 2015.

[24]. Bashir Yusuf Bichi , Tuncay Ercan , Anans Mu'azukademi , *"An Efficient Resource Management in Cloud Computing"* , International Conference on Advanced Technology and Sciences(ICAT) , 2016.

[25]. Dr. Amit Agarwal , Saloni Jain , *"Efficient optimal algorithm of task scheduling in cloud computing environment"*, International Journal of Computer Trends and Technology(IJCTT),2014.

[26]. Bhavisha Kanani , Bhumi Maniyar , *"Review on Max-Min Task Scheduling Algorithm for Cloud Computing"* , Journal of Emerging Technologies and Innovative Research (JETIR) , 2005.

[27]. Wang, Lizhe, *" Cloud computing: methodology, systems, and applications"*, CRC Press, 2012

[28]. Thomas, A., Krishnalal, G., Raj, V. J. , *"Credit based scheduling algorithm in cloud computing environment",* Procedia Computer Science. Vol. 46, 2015.

[29]. Bao-Ngoc Trinh, Quyet-Thang Huynh, Xuan-Thang Nguyen, Van-Quyen Ngo, Thanh-Trung Vu, *"Application of nash equilibrium based approach in solving the risk responses conflicts"*, Journal of Science and Technology - Le Quy Don University, No. 193 (10-2018).

[30]. Tek4.vn, *"Điện toán đám mây là gì ?"*, Điện toán đám mây là gì? (tek4.vn), Jan 1. 04, 2022.

[33]. Pendharkar, P. C.: *"Game theoretical applications for multi-agent systems"*, Expert Systems with Applications, 2012, 39, (1), pp. 273-279.

[34]. Siar, H., Kiani, K., and Chronopoulos, A. T.: *"An effective game-theoretic static load balancing applied to distributed computing"*, Cluster Computing, 2015, pp. 1-15.

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

[36] A Game Theoretic Method for Resource Allocation in Scientific Cloud January 2016 International Journal of Cloud Applications and Computing 6(1):15-41 DOI:10.4018/IJCAC.2016010102

[37] Trinh Bao Ngoc, "*ÁP DỤNG LÝ THUYẾT TRÒ CHƠI VÀ CÂN BẰNG NASH XÂY DỰNG PHƯƠNG PHÁP MÔ HÌNH HÓA XUNG ĐỘT TRONG QUẢN LÝ DỰ ÁN ĐẦU TƯ CÔNG NGHỆ THÔNG TIN VÀ THỬ NGHIỆM TRONG MỘT SỐ BÀI TOÁN ĐIỂN HÌNH"*, Hanoi 2020

[38] SM. Mousavi and G. Fazekas: A Novel Algorithm for Load Balancing using HBA and ACO in Cloud Computing Environment, International Journal of Computer Science and Information Security, Vol. 16, 2016, pp. 48-52

[39] D. Ye and J. Chen, “Non-cooperative games on multidimensional resource allocation,” Future Generation Computer Systems, vol. 29, no. 6, 2013

[40] Diptangshu Pandit, Samiran, Matangini Chattopadhyay, Nabendu Chaki , Resource Allocation in Cloud using Simulated Annealing, 2014 Applications and Innovations in Mobile Computing (AIMoC)

[41] Boxing Tan, Hui Ma, Yi Mei, A NSGA-II-based approach for service allocation in cloud, Victoria University of Wellington, 2017

[42] A. Paschke and E. Schnappinger-Gerull, “A categorization scheme for sla metrics.” Service Oriented Electronic Commerce, vol. 80, no. 25-40, p. 14, 2006

[43] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,” Future generation computer systems, vol. 28, no. 5, pp. 755–768, 2012.

[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, Apr 2002