Dynamic Resource Allocation in Cloud Computing Using a Combination of Meta-heuristic Algorithms

Shadi Samieifar[1], Farhad Mardukhi[2, 3]
Department of Computer Engineering, Faculty of Engineering, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran

Department of Computer Engineering, Faculty of Engineering, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran
Department of Computer Engineering and Information Technology, Faculty of Engineering, Razi University, Kermanshah, Iran

Abstract
Cloud computing can be considered one of the largest IT solutions to industry problems. In fact, this concept can be used for features such as processing power or storage service provider cloud only through the Internet connection costs low without worrying about hardware upgrades or software updates used software. But because of a cloud may not be responsive to user needs, recently introduced a new concept called association clouds. The concept is to meet the needs of users, a few clouds at a time and uses integrated management. On the other hand due to the increasing use of cloud computing technology providers are faced with new challenges that one of the most important scheduling and allocation of resources. The problem is that the NP-hard, there is no capacity to respond to it in a reasonable time using linear methods. So in this study, a metaheuristic method based on a combination of genetic algorithms and colonial competition used. Since these two algorithms have shown a good performance in this area, in this study, the two algorithms are combined in series with the synchronous target to reduce implementation time and costs were considered. Finally, simulations show that the proposed method has better answers to these two algorithms has produced.
Keywords:
Cloud computing, resource allocation, scheduling, community clouds, genetic algorithm, Imperialist Competitive Algorithm

1. Introduction

One of the most ironic things is that although IT is very high computing power, but it seems to work with slower computers and computer speed is not old! The reason for this is that every day are complex operating systems and software packages; but today we will not be worried about using cloud computing technology, because we keep using the idea of cloud computing it on a server on the Internet and from anywhere in the world using a simple personal computer, we pay to process the increasing needs of users with different sources of cloud computing as one of the technology, now based services to rapidly progress. Providers of cloud computing services, an expensive software that requires high computational power and memory through the clouds by the clients are high. It has all the resources and the various components of cloud service providers and end customers need to notify the specialty and pay not install this kind of service, so customers pay only for the service, due to this, it is clear that cloud-based services are one of the most popular services on the Internet have become, but these services have different challenges such as security and privacy, costs and delivery, portability, reliability and availability, efficiency and cost of bandwidth. In order to better understand cloud computing, it is better to be familiar with a variety of cloud computing. In general, cloud computing can be divided into two models:
Deployment models: means location and management of computing infrastructure
Service models: including different types of services that you can use the cloud computing platform Benefits of cloud computing include

Manuscript received April 5, 2017
Manuscript revised April 20, 2017

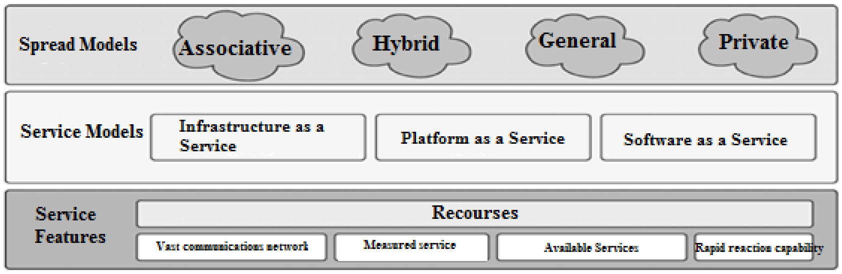


Figure 1 - Definition of cloud computing from the perspective of NIST [4] institute

Easy Management
With this technology, the IT professionals spend less time and energy to manage cloud computing system.

Reduction in costs
The most important advantages of cloud computing to reduce costs. With this technology there is no need to upgrade software or hardware.

Uninterrupted service
According to the like-stop service Gmail in 2009, as well as cloud services Amazon in the same period, but cloud services than traditional systems, can respond to user needs without interruption.

Clean processing
Without the use of cloud computing will have on all nodes in an organization or even home users, volume and computational processing power and unnecessary at the same time that this has been the carbon dioxide and consumes large amounts of electrical energy from the other side as unnecessary but the use of cloud computing in addition to reducing carbon dioxide emissions and reduce greenhouse gases, it also reduces electrical energy consumption.

There are also challenges in cloud computing is that it challenges

Security
Many believe that if we are on our side and we have monitored, will determine the heightened security; it should be noted, however, that contrary to the impression, the use of cloud computing can enhance the security of information.

Privacy
User data storage in cloud computing environments, it evokes the idea in mind how one can be sure of maintaining the confidentiality of his/her data? Users' privacy in the cloud servers, both the government and the strikers is verifiable. Cloud computing mechanisms and policies should be such that always guarantee the confidentiality of your account information.

Reliability
So far, many policies to enhance reliability of cloud computing environment has been presented. The goal of all these methods avoid cutting implement user services and to achieve the final result is accurate and reliable.

Suggested Method
In this section we will discuss the proposed approach. First, we will offer it separately with genetic algorithms and colonial competition, then we present our proposed approach in the form of a combination of the two algorithms. The proposed method is taken into account on a community cloud is a collection of several public and private cloud. Today various organizations in addition to having their own private clouds that are commonly used either for storage or processing power are not many, of one or more public cloud use. So the focus of this study, hybrid and community clouds on the set of clouds.

Problem solving using genetic algorithms
A genetic algorithm metaheuristic that has shown very good performance for various problems. This algorithm needs to produce original responses that generally randomly generated.

How to show answer
In all metaheuristic, because of the need to produce an answer, at first, be sure answers to that structure in accordance with the specified structure, called how to show answer. Here we use an array structure for how to show answer. Genetic algorithms in this array, called chromosomes. A one-dimensional optimization problem, N_{var} chromosome, $1\times N_{var}$ is an array. This array is defined as follows.

1- Choromosome $[P_1 .P_2 .P_3 ....P_{Nvar}]$

Accordingly, each answered an array of 1 in N_{var}, where N_{var} is the number of tasks in the queue to run and the arrays it represents the number of virtual machines that are assigned to run it. For example, in Figure 3-2, the task of hosting No. 1 in No. 3 and No. 5 in a virtual machine (7) has been tasked with assuming that the number of virtual machines is at least 7.

3	1	4	6	7	2	2	1

Figure 2 – Scheme of show answer

Production of basic answer

In the proposed algorithm, the initial response strategy is random. Accordingly, as shown in Figure 3-2 randomly created chromosomes that element in it will be numbers between 1 to the number of virtual machines.

Selection

Parents choose to produce new offspring follow different procedures. After calculating the probability of each feature based on their performance, sampling discrete distribution roulette wheel is used to select the next feature. Roulette wheel selection which was first introduced by Holland is one of the most convenient methods of random selection that the idea of the likely choice. The probability of selecting each pass can be obtained from the following relationship.

$$2 - P_i = \frac{Q_i}{\sum_{i=1}^{n} Q_i}$$

Where Q_i is the i call quality. The roulette wheel sector represents the probability of selection for each of the answers will be. Roulette wheel in this way acts to select each feature, a random number between one and zero in each interval of production and number were chosen the corresponding call. Implementation methodology roulette wheel in this case is that a circle is considered and it is divided into a number of replies so that each section corresponds to the probability of selection for each call. Rotate the wheel (random number generator), and where the wheel stopped to look wheel indicator, the answer is related to that selection.

Intersection

Search space takes place by generating new chromosome from chromosome earlier. Operator intersection of two chromosomes randomly selected, with probability and operations over the strings makes new chromosomes, so that the structural characteristics of parents and children combined. We know that in each iteration of the genetic algorithm, parents are produced in the population should be selected and combined with each other in order to come to a new population. To do this, you need an operator that certain compounds in this study, the combination of a single-point, two-point composition and mixing is considered. The combination of a point, the point is randomly selected chromosome range. The first parent and the parent has transferred a part of their second child, another part of it. So eventually the children will be similar to Figure 3-3.

Parent

3	1	4	6	7	2	2	1
7	5	1	3	6	4	2	4

Children

3	1	4	3	6	4	2	4
7	5	1	6	7	2	2	1

Figure 3 - Single point

There is another combination of such methods and integrated approach two spot due to its simplicity and good performance combine single point, this method is used.

Mutation

For a variety of coding systems for chromosomes, different operators used for jumps. On one or more chromosomal genes mutation operator with a certain probability that a randomly chosen to change and change its value. This function as a basic function is to protect the population density is considered, so that new parts of the space that operators choose not to access and intersection also be sought. Mutations in the process of our algorithm that is how random a gene (virtual machine is allocated to a task) has been selected and were randomly assigned to one of the other possible values will change. Obviously, this value should be one of the possible values for cars and produced the answer is justified. In the absence of justification, a new answer is produced.

Objective function

Based on what has already been said, the issue is intended for the purpose intended. In fact, the objective function contains two actions will be optimized:

- ***Reduce the total make span time***

The running time is defined tasks in a multi-processor system or parallel. This value is not measurable before it is implemented. But the following calculations can be largely traced it. The variance is calculated along the lines created for virtual machines. Obviously, the lower the variance, how the distribution of tasks among virtual machines have been better. So we have:

$$3\text{-}var = \sqrt{E[(RelativeRatio - \mu)^2]}$$

Where Relative Ratio relative length of the lines created for each virtual machine, μ and E the expected average length of these lines are shown. As the procession is considered a suitable criterion alone cannot be taken into consideration is the relative length divided by the length of the queue on the virtual machine's processing power. The relative length obtained ranks the ability of virtual machines is also considered. Since the cloud is intended for a community cloud, the ability of virtual machines also will be different. So we have:

$$4 - RelativeRatio_i = \frac{QeueLength_i}{VMPower_i}$$

Where Relative Ratio relative length of the queue for virtual machine i have made up, Queue Length actual queue length for virtual machine i have produced and directed by millions of virtual machine i have in terms of processing power VM Power million instructions per second.

Cost of compliance

In order to estimate the cost of using virtual machines, you must first estimate the execution time of each virtual machine. Then be multiplied by the Price for virtual machines. So we have:

$$5 - Cost_i = \left[\frac{QeueLength_i}{VMPower_i * 3600}\right] * HourlyCost_i$$

Where Hourly Cost per hour is the use of virtual machine type i.
In fact, the objective function will be as follows:

6 - f = Minimize (Cost. Make spone)

Since the function intended target was a double objective, Topsis should be used to rank replies. About Topsis discussed in detail in Chapter 2. Must now consider the coefficient for each of the goals, determine their impact on the scoreboard. In this study, experiments, this value is equal to 0.5 were considered both goals.

Stopping criterion

Various criteria is to stop an algorithm. In this study, in order to simplify, the number of repetitions determined to stop the algorithm is used.

Elements of Genetic Algorithm

Elements algorithm	Uses strategies taken
Answer	A presentation with the right amount
The initial answer	Random feasible solution with random evolutionary algorithm
The solution space	Given the limits of the feasible solution
Evaluation Criteria	Objective function
Choose the best neighborhood	Roulette wheel
Strengthening Strategy	Combined action
Diversification strategy	Practice jumps
Stopping criterion	The maximum number of repetitions

Problem solving using ICA, like other evolutionary algorithms, this algorithm, the initial random number with which each of them is a "country" called, begins. Some of the best elements of the population (equivalent elite GA) elected as imperialist. The remaining population also as a colony, are considered. Depending on the power colonists, the colonists with a special process that follows, to kill himself. The total power of the empire, both forming part of the imperialist country (for the core) and its colonies, it depends. In math mode, this dependence with imperial power is defined as the sum of the Imperial power, plus a percentage of the average colonial power, is modeled. With the formation of the early empires, imperial competition between them begins. Every colonial empire cannot compete, succeed and increase their power (or at least reduce the influence stop), the colonial competition, will be removed. Therefore, the survival of an empire depends on its ability to attract rival colonial empires, and the domination they would be out. Consequently, in the imperialist competition, gradually added to the power of larger empires and kingdoms and the weaker will be removed. Empires to increase their power, would be forced to develop their own colonies. Over time, the colonies, in terms of power and control to be closer empires will be a kind of convergence. The end of colonial competition, when a single empire in the world, we matched the colonial position, your imperialist country, are very close.

Forming early empires

In optimization, the aim of finding an optimal solution in terms of the variables of the problem. We create an array of variables that must be optimized problem. GA in this array, called chromosomes. Here, too, we call it a country. N_{var} is one-dimensional optimization problem, a country, $1\times N_{var}$ is an array. This array is defined as follows.

7- $country=[p_1, p_2, p_3,..., p_{N_{var}}]$

Variable values in one country, for integers between 1 and the number of virtual machines are displayed. From the perspective of cultural history, social features can be components of a country - political, such as culture, language, economic structure and other characteristics are taken into account. Figure 3-6 illustrates this problem. According to this form of unknown variable cost function in the optimization process we're looking for them, socio-political, historical and cultural characteristics are at the low point lead the country to make a cost function. In fact, solving an optimization problem introduced by the algorithm, we are looking for the best country (the country with the best of social and political). Find the country is in fact equivalent to finding the best parameters that produce the lowest cost function.

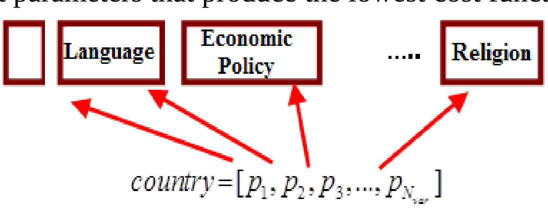


Figure 4 - Socio-political components forming a country

In this study, we want virtual machines to be assigned to different requests entered into the system the costs recorded in its objective function is minimal. In a typical case, answers for those questions that might lead to a stable output can be defined. Initial set of answers to this problem as we could. In this country to be defined.

8 - $country=[p_1, p_2, p_3,..., p_{N_{var}}]$

To start the algorithm have a number of these countries (number of countries algorithm) be created. The initial matrix is formed randomly throughout the country. The cost of one country to evaluate the function f will be as follows:

9- $cost_i = f(country_i) = f(p_1, p_2, p_3,..., p_{N_{var}})$

$f(country_i) = minimize(Makespan.Cost)$

In the present case, the answer is an array of 1 in N_{var} where N_{var} number of tasks available, for example, in Figure 3-7, task No. 1 at home (3) and (7) has been the task of No. 5 in a virtual machine.

3	1	4	6	7	2	2	1

Figure 5 - Scheme of show answer

Production of basic responses

To start the algorithm, we make an initial number of $N_{country}$ country. N_{imp} to the members of this population (Countries with the lowest cost function) is picked as the imperialists. The remaining N_{col} to the countries that constitute belong either to a colonial empire. For the first inter-imperialist division of the colonies, to any imperialist number of colonies that these numbers, it is a powerful fit. To do this, with the cost of all the imperialists, in the format they consider normalized cost.

10 - $C_n = \max_i\{c_i\} - c_n$

Where c_n, expenses imperialists n, $\max_i\{c_i\}$ the highest cost among the imperialists and C_n, normalized cost of the imperialists, Every imperial war that has cost more (imperialism is weaker), with normalized cost will be less. With the cost normalized, normalized relative strength of the imperialist calculated as follows, based on which, the colonial countries, the imperialists divided.

11-

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right|$$

From a different perspective, normalized power in an imperialist, colonial ratio by which the imperialists run. The initial number will be equal to an imperialist colonies

12- $N.C._n = round\{p_n.(N_{col})\}$

Where $N.C._n$, the number of colonies of the empire and N_{col} the total population of the countries colonized in the primary. *round* function, which is also the nearest whole number to a decimal number. Considering $N.C.$ for each empire, to randomly select the number of countries in the early colonial and imperialist our n. With the initial state of all empires, ICA begins. To meet the latest evolution in a ring that a stop condition will continue.

Displacement of colonial and imperialist position

At the same socio-political policy of attracting the destruction of the colony in some cases lead to positive results for them. Some countries as a result of this policy achieved some kind of public self-confidence and after a while the educated (in other words absorb those colonial culture) were who have led their nation to escape from the clutches of colonialism. During the colonial colonies in the country, some of these colonies may be in a better position than the imperialists (To the point that they cost less than the cost function value of the cost function in the position of imperialist produce.) In this case, the country colonizer and colonized country, where it changes the algorithm together with the new situation has continued colonial country and this time the country is starting to apply the new imperial colonial policy of assimilation in itself. Changing a colonizer and colonized, shown in Figure 3-11. In this way, the imperial colony, which cost less than your imperialist, a darker color, is shown. Figure 3-12 shows the entire empire after changing position.

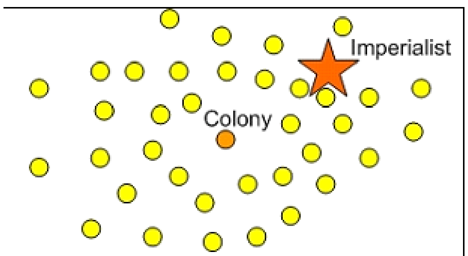


Figure Error! No text of specified style in document. The whole empire, after changing situations

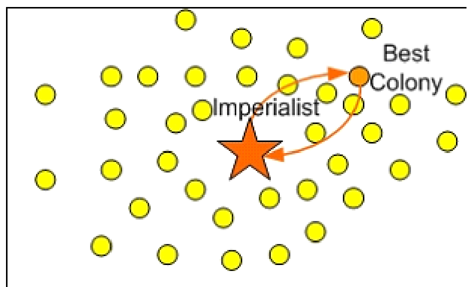


Figure 6 - Changing a colonizer and colonized

Total Power of an empire

Empire is a powerful country against colonial power, plus the power of its colonies. Thus, for the cost of an empire, we have.

13- $T.C._n = Cost(imperialist_n) + \xi\, mean\{Cost(colonies\ of\ empire_n)\}$

Where $T.C._n$ cost of empire n and ξ is a positive number, usually between zero and one and is considered close to zero. Little consideration of ξ, causes the total cost of an empire, almost equal to the cost of the central government (Imperial), and increased ξ also increased the effect of the cost of a colonial empire in determining the total cost. In a $\xi = 0.05$ in most cases implementation has led to good results in this study is taken into account.

Competitive Imperialist

As previously stated, any empire which can increase the power and strength of your competition lose, in the imperialist competition, will be removed. Uninstall is done gradually. This means that over time, empires weak, lost its colonies and empires stronger, it has acquired colonies and adds his own power. To model this reality, we assume that we removed empire, the empire is weakest. Thus, the repetition of algorithms, one or more of the weakest colonial empire has taken to seize the colonies, we create competition between all empires. Colonies are, not necessarily by the most powerful empire will not seize, but more powerful empires, are more likely takeover. Figure 3-13 shows the overview of this section of the algorithm.

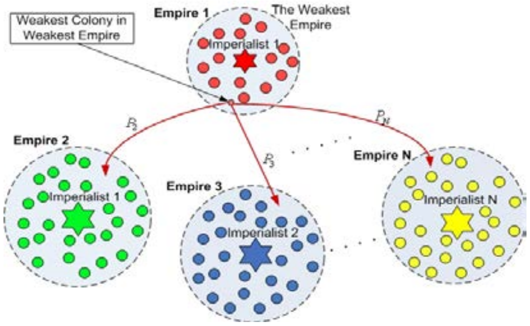


Figure 8 - The overarching theme of colonial competition: larger empires, the more likely they are to take other colonial empires.

In this form (1) Empire considered as the weakest taken and subjected to imperialist rivalry has been one of its colonies and empires 2 to N are competing to seize it. For modeling competition for the appropriation of the colonial empires, the possibility of a takeover of Empire (which is proportional powerful empire), and taking into account the total cost of empire, we calculated as follows. The total cost of the empire, the total cost of our targeted normalized.

14- $N.T.C._n = \max_i\{T.C._i\} - T.C._n$

In this regard, $T.C._n$, total cost of empire n and $N.T.C._n$, the total cost of that empire is normalized. Every empire have less $T.C._n$ will be more. In fact, the cost of an empire $N.T.C._n$ and $T.C._n$ is equal to total power. Empire at the lowest cost, highest power. With a total cost of normalized probability (power) takeover colony of competition, by any empire, is calculated as follows.

15- $$p_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \right|$$

With a chance to seize the empire, such as Roulette Wheel mechanism in the genetic algorithm is needed to powerful colonial empires in competition with probability proportional to one of them. With a chance to seize the empire, because these colonies randomly, but probably affiliated with a chance to seize the empire, divided between empires; the vector P from the above probability values, we are following.

16- $$\mathbf{P} = \left[p_{p_1}, p_{p_2}, p_{p_3}, \ldots, p_{p_{N_{imp}}} \right]$$

Vector P the size is 1*N_{imp} of probability values appropriation is made up of empires. Then the random vector R, we were the same size as P vector. This vector arrays, random numbers with uniform distribution in the interval [0,1]. Vector P has size and is made up of empires takeover of probability values. Then the random vector R, we were the same size as P vector. This vector arrays, random numbers with uniform distribution in the interval [0,1].

17- $$\mathbf{R} = \left[r_1, r_2, r_3, \ldots, r_{N_{imp}} \right]$$

$$r_1, r_2, r_3, \ldots, r_{N_{imp}} \quad U(0,1)$$

Then we have formed our vector D as follows.

$$\mathbf{D} = \mathbf{P} - \mathbf{R} = \left[D_1, D_2, D_3, \ldots, D_{N_{imp}} \right]$$

18 $$= \left[p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3} - r_3, \ldots, p_{p_{N_{imp}}} - r_{N_{imp}} \right]$$

With vector D, colonial empire that we mentioned about it in the vector index D is larger than the rest. Empire most likely have possessed more likely to index the vector D, will be the highest value. No need to calculate the CDF makes this mechanism works much more quickly than the roulette wheel. The new mechanism proposed to allocate colonial empire not only can they be useful in terms of the possibility of takeover, but as a selection mechanism based on the possibility of replacing the roulette wheel in the genetic algorithm to choose the parents and it greatly increases the speed of operations. With the acquisition of colonial empires by one of the operations of this stage of the algorithm is over.

Fall of weak empires

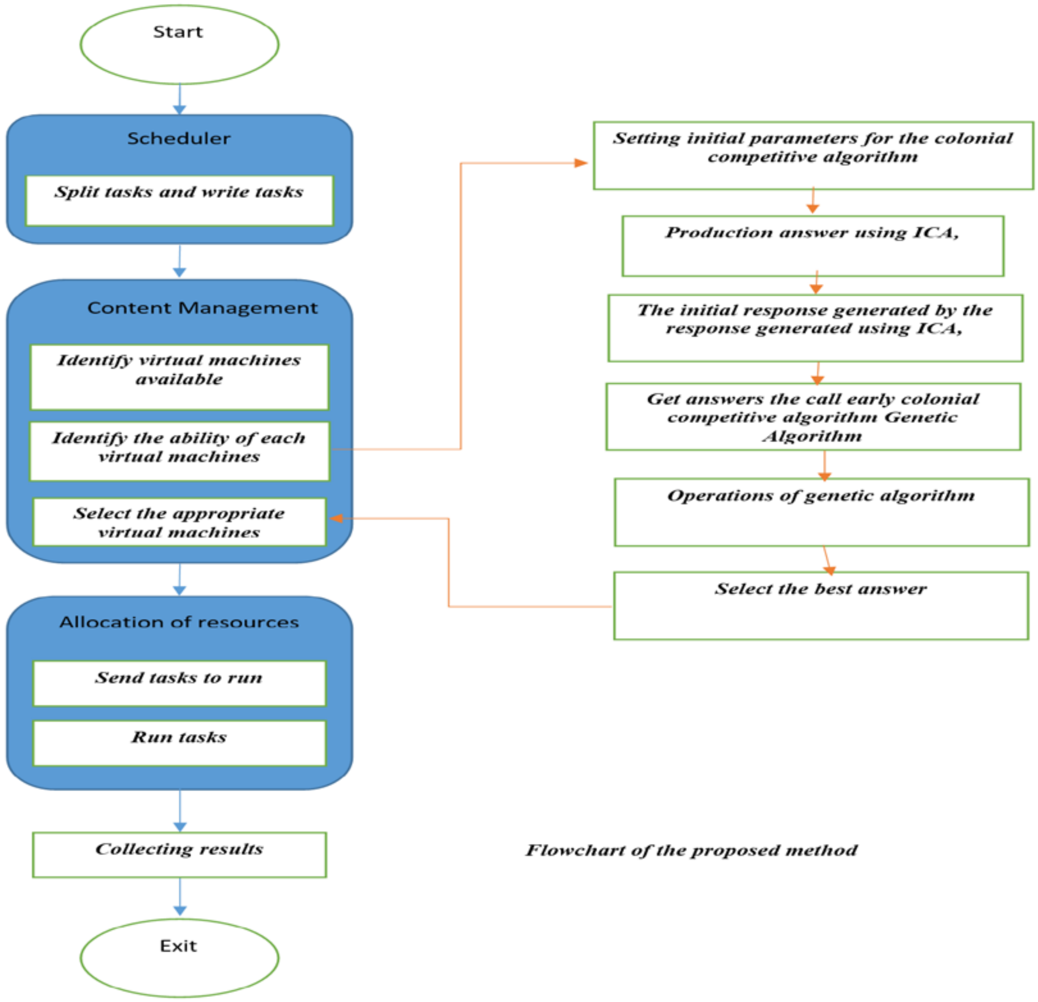
During the imperialist competition, inevitably, weak empires and colonies gradually fell into the hands of powerful empires fall. Different conditions can be considered for fall of an empire. In the proposed algorithm has since been removed is considered an empire that has lost its colonies. An algorithm is intended to meet the convergence criteria, or until the total number of iterations, continues. After a while, all empires have fallen and will have only an empire, and the rest under the control of a single empire, they argue. The new ideal world, all settlements, run by an imperial units. Position and costs colonial, imperialist country is equal to the opportunity cost. In this new world, the differences, not only between colonies, but there is the colonial and imperialist country. In other words, all countries, at the same time, the colonies and their colonizers. In such a situation as a condition of imperialist rivalry over and stop the algorithm stops. The proposed methodology using colonial competition similar to the method using a genetic algorithm is proposed. At first things are registered in the system. Then the scheduler are given. Scheduling tasks and Content Manager is broken into small tasks. Content Manager Task of identifying the available virtual machines and the ability of each of them is responsible. After the virtual machine characteristics and properties of each of the tasks, whether the commands were recognized and the amount of resources needed, an Algorithm for appropriate allocation of resources should be used. Here imperialist competitive algorithm is used. The imperialist competitive algorithm is initialized after the initial responses are randomly generated. Then the colonizers and the colonized features they bow their acquisition deviation. After these changes in acceptance call using the objective function is calculated and if fitted to each of these colonies was higher than the corresponding colonial, rather than changing them. Then the total cost of an empire is calculated by taking the colonies. Then the weakest of the weak colony is selected and its empire colonialism colony to join the most similar to it. Finally, it should be checked whether or not the transfer is empire was empty. If there is no colonial empire, the empire will be removed. This procedure continues until the stop condition is established. Finally, the product of imperialist competitive algorithm is applied, is an appropriate response to how the allocation of resources as far as possible has set the objective function in terms of performance is at an appropriate level. The proper response produced as a response indicating that each virtual machine is allocated to each task is presented. Finally, the call to the Resource Manager to use this approach to resource allocation. Finally tasks have to be performed to the corresponding virtual machines.

Elements colonial competitive algorithm

Elements algorithm	Uses strategies taken
Answer	A presentation with the right amount
The initial answer	Random feasible solution with random evolutionary algorithm
The solution space	Given the limits of the feasible solution
Evaluation Criteria	Objective function
Choose the best neighborhood	Roulette wheel
Strengthening Strategy	Assimilation policy
Strategy to diversify	Revolutionary action
Stopping criterion	The maximum number of repetitions

The proposed method combines

The proposed method combines is for the series. Thus, in the Algorithm competition colonial answers are generated, then call near its optimal on genetic algorithm has genetic algorithms perform random answers first, of appropriate responses generated imperialist competitive algorithm used them.



Flowchart of the proposed method

Findings

For this study, the application Netbeans Version 8.1 is used. The Java Development Kit software Version 8 and CloudSim Version 3 has been used. Also this software on a system with Intel CoreI7 6500U processor with 8GB of RAM is implemented. Algorithms presented in this study and previous algorithms have been implemented in this application.

Results

Parameters

Parameters intended for issue

Table 1 - The parameters of the problem

The number of service providers	The number of virtual machines	The total number of tasks
4	130	40315

Table 2 - Parameters of virtual machines is considered

Number	Main memory (MB)	Processors (million instructions per second)	Service Provider
30	1000	512	Amazon
20	1000	1000	IBM
50	1000	2000	Azure
30	2000	1000	Rack Space

- Genetic algorithm parameters

Table 3 - Parameters of Genetic Algorithm

Number of chromosomes	Number of generations	Compound rate	Mutation rate	Mutation percent
50	100	0.8	0.2	0.2

- Colonial competitive algorithm parameters

Table 4. Parameters colonial competitive algorithm

Number of countries	Number of colonists	Number of decades	Rate of revolution	Threshold adopted
50	10	1000	0.2	0.02

- Parameters of combined method

Table 5 - The parameters of combined method

Number of countries	Number of colonists	Number of decades	Rate of revolution	Threshold adopted	Number of chromosomes	Number of generations	Mutation rate	Compound rate
50	10	500	0.2	0.02	50	50	0.2	0.8

Results and Comparison

Due to the random nature of meta-heuristic algorithms, comparing them should be done in several consecutive repeating the average of the values is detected. The results of 10 repetitions with the average for the time of implementation tasks are shown in Table 4-6.

Table 6 - Values obtained for runtime in 10 consecutive performances

10	9	8	7	6	5	4	3	2	1	
17835.1	18814.28	17835.1	18211.91	18814.28	18260.44	18463.51	18121.74	18753.96	18334.41	GA
19991.32	19581.36	18185.14	18694.7	18264.53	18191.78	19991.32	18237.26	18309.87	18328.69	ICA
13465.99	13364.21	12817.71	13364.21	13465.99	13364.21	12817.71	12640.71	13364.21	17968.23	ICA_GA

Based on what is specified in Table 4-6, the values obtained in repeated every 10 values obtained for the proposed method of GA and ICA algorithm is improving own again. According to Figure 4-1 is clear that in the 10 run, always amounts obtained for the proposed method can be significantly lower than the values obtained for the two algorithms are GA and ICA.

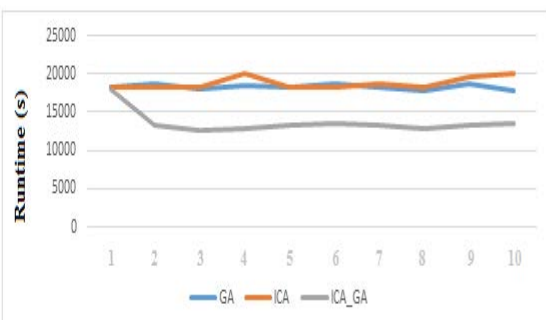


Figure 9 - Values obtained for the time run in 10 consecutive performances

Based on what is known in Figure 4-1, the amount of runtime in principle for the proposed method is significantly less than GA and ICA algorithms. Finally, the average time of implementation of these algorithms is shown in Figure 4-2. According to this figure, the proposed method is much better than an average execution time is two genetic algorithms and colonial competition. The total running time of 18344 seconds and 18777 Genetic Algorithm in colonial competitive algorithm, the proposed procedure is 13663 seconds.

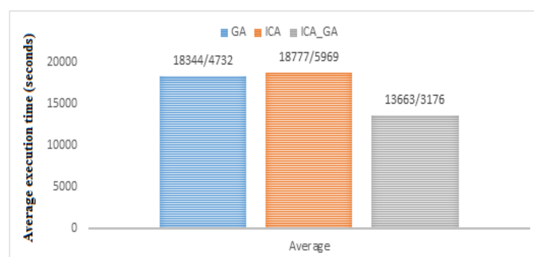


Figure 10 - The average values obtained for the fifth time in a row enforcement

The objectives of this study, reducing the running costs of the user who also modeled in the form of objective function. Cost values obtained for 10 consecutive run is shown.

Table 7 - Average cost for runtime obtained in 10 consecutive performances

10	9	8	7	6	5	4	3	2	1	
17835.1	18814.28	17835.1	18211.91	18814.28	18260.44	18463.51	18121.74	18753.96	18334.41	GA
19991.32	19581.36	18185.14	18694.7	18264.53	18191.78	19991.32	18237.26	18309.87	18328.69	ICA
13465.99	13364.21	12817.71	13364.21	13465.99	13364.21	12817.71	12640.71	13364.21	17968.23	ICA_GA

Based on what is shown, the proposed method is almost always less for a fee. This gap, however, does not seem like a significant total running time, but improvement has been achieved. The average cost in dollar terms in 10 consecutive performances. The proposed method is able during the run about $ 28 to about $ 49 genetic algorithms and improvements to the colonial competitive algorithm.

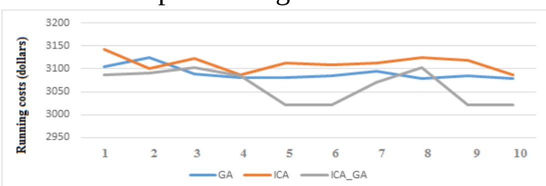


Figure 12 - Values obtained for a fee of 10 consecutive performances

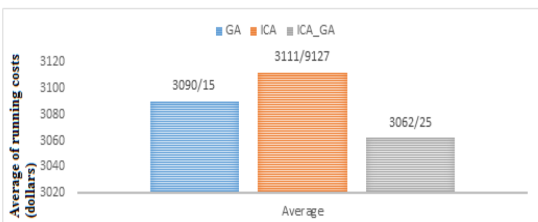


Figure 13 - The average values obtained for a fee of 10 consecutive performances

Also, compare the costs derived based on worst case is shown in Figure 4-6 based on what we can see in Figure 4-6, the proposed method significantly better at worst, it has produced responses.

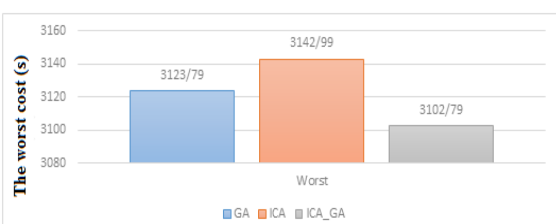


Figure 14 - The worst cost obtained for 10 runs

On the other hand the algorithm should be examined overhead to the runtime is also in good condition. Accordingly, the time the 10 simulation runs are shown in Table 4-8. According to what is shown in Figure 4-7, time simulation in a state average of 44 and 49 seconds to 36 seconds for genetic algorithm and colonial rivalry reached on the proposed approach show that the proposed algorithm in addition to reducing total execution time and costs, has been able to show appropriate responses in a reasonable time and even less genetic algorithms and colonial competition.

Table 8 - Values obtained for a fee of 10 consecutive performances

10	9	8	7	6	5	4	3	2	1	
44	41	45	44	42	43	43	44	45	46	GA
44	42	41	42	41	39	38	38	42	38	ICA
33	39	33	39	36	33	37	36	37	38	ICA_GA

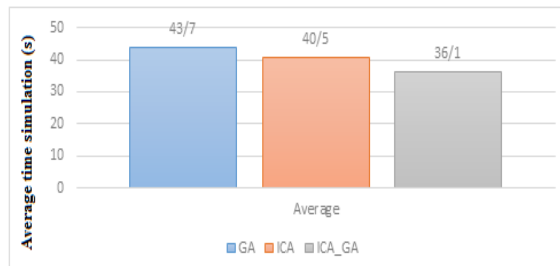


Figure 15 - Average time simulation obtained for 10 runs

The worst-case values shown in Figure 4-8 is clearly derived values that the proposed method is less than methods based on genetic algorithms and colonial competition.

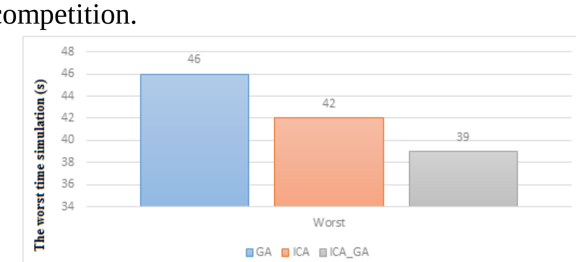


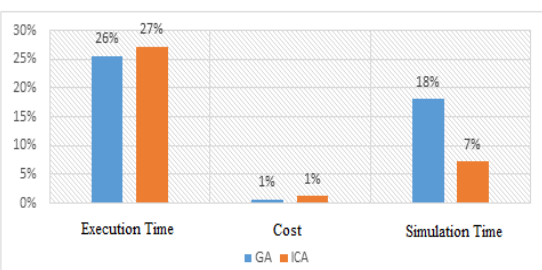
Figure 16 - The worst time simulation obtained for 10 runs

Conclusion and Summary

The main focus of this study is to provide a meta-heuristic method for allocating resources in the cloud computing community by reducing runtime and user fees have been reduced. However, it seems that these two objectives are not always in a direction, but through the rankings suitable answers using the Pareto front and Topsis was possible. Based on what was seen, the proposed method has failed both in terms of overall execution time and cost to improve ties compared to methods based on genetic algorithms and algorithms colonial competition. It is also due to lower overhead and are able to produce the answers in less time by comparing the proposed method can be downloaded using the proposed method based on genetic algorithm running time of less genetic algorithms have been compared to the amount equal to 26 percent. Also, using the proposed method, user fees also decreased about 1 percent from a genetic algorithm is own again. The proposed algorithm produced less time and your answers in other words, a reduction of 18% in the simulation, the algorithm also lowered overhead. On the other hand, by comparing the proposed method can be downloaded using the proposed method based on colonial competitive algorithm execution time of less compared to colonial competitive algorithm has been compared to the amount equal to 27 percent. Also, using the proposed method, user fees are about 1% from colonial competitive algorithm has declined. The proposed algorithm has produced in less time and your answers in other words, with a 7 percent decrease simulation time, reducing overhead algorithm.

Table 9 - The recovery rate of the proposed method

	Execution time	Cost	Simulation Time
GA	26%	%	18%
ICA	27%	%	7%



Proposed Future Works

As future work, we are trying to change the model, the objective function in ways that are cost effective and better way. Also a comprehensive study, we will discuss the types of algorithms are presented and compared with them.

References

[1] R. L. Grossman, "The case for cloud computing," IT professional, vol. 11, no. 2, pp. 23-27, 2009.
[2] Y. Jadeja, and K. Modi, "Cloud computing-concepts, architecture and challenges." pp. 877-880.
[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of internet services and applications, vol. 1, no. 1, pp. 7-18, 2010.
[4] B. Sosinsky, Cloud computing bible: John Wiley & Sons, 2010.
[5] J. W. Rittinghouse, and J. F. Ransome, Cloud computing: implementation, management, and security: CRC press, 2016.
[6] J.-F. Li, and J. Peng, "Task scheduling algorithm based on improved genetic algorithm in cloud computing environment," Jisuanji Yingyong/ Journal of Computer Applications, vol. 31, no. 1, pp. 184-186, 2011.
[7] Y. Ge, and G. Wei, "GA-based task scheduler for the cloud computing systems." pp. 181-186.
[8] L. Liu, M. Zhang, R. Buyya, and Q. Fan, "Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing," Concurrency and Computation: Practice and Experience, 2016.
[9] H. Seidgar, M. Kiani, M. Abedi, and H. Fazlollahtabar, "An efficient imperialist competitive algorithm for scheduling in the two-stage assembly flow shop problem," International Journal of Production Research, vol. 52, no. 4, pp. 1240-1256, 2014.
[10] J. Geelan, "Twenty-one experts define cloud computing," Cloud Computing Journal, vol. 2, pp. 1-5, 2009.
[11] P. Mell, and T. Grance, "The NIST definition of cloud computing," 2011.
[12] M.-G. Avram, "Advantages and challenges of adopting cloud computing from an enterprise perspective," Procedia Technology, vol. 12, pp. 529-534, 2014.
[13] B. Grobauer, and T. Schreck, "Towards incident handling in the cloud: challenges and approaches." pp. 77-86.
[14] H. Huang, H.-B. Xie, J.-Y. Guo, and H.-J. Chen, "Ant colony optimization-based feature selection method for surface electromyography signals classification," Computers in biology and medicine, vol. 42, no. 1, pp. 30-38, 2012.
[15] B. Liu, M. Tian, C. Zhang, and X. Li, "Discrete Biogeography Based Optimization for Feature Selection in Molecular Signatures," Molecular Informatics, vol. 34, no. 4, pp. 197-215, 2015.
[16] N. Hordri, S. Yuhaniz, and D. Nasien, "A Comparison Study of Biogeography based Optimization for Optimization Problems," Int. J. Advance. Soft Comput. Appl, vol. 5, no. 1, 2013.
[17] M. Tahmassebpour, A. M. Otaghvari, "Increase Efficiency Data Processing With Using an Adaptable Routing Protocol on Cloud in Wireless Sensor Networks," Journal of Fundamental and Applied Sciences, vol. 8, no. 3S, pp. 2434-2442, 2016.
[18] J. Pearl, "Heuristics: intelligent search strategies for computer problem solving," 1984.

[19] S. F. Rodiyansyah, “The Technique For Order of Preference By Similarity To Ideal Solution (TOPSIS) untuk Penentuan Penerima Beasiswa,” Infotech Journal, vol. 1, no. 1, 2016.
[20] S. Opricovic, and G.-H. Tzeng, “Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS,” European journal of operational research, vol. 156, no. 2, pp. 445-455, 2004.
[21] J.-J. Huang, and K. Yoon, Multiple attribute decision making: methods and applications: Chapman and Hall/CRC, 2011.
[22] U. Schwiegelshohn, and R. Yahyapour, "Analysis of first-come-first-serve parallel job scheduling." pp. 629-638.
[23] F. Bonomi, “On job assignment for a parallel system of processor sharing queues,” IEEE Transactions on Computers, vol. 39, no. 7, pp. 858-869, 1990.
[24] P. Kumar, and A. Verma, "Scheduling using improved genetic algorithm in cloud computing for independent tasks." pp. 137-142.
[25] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, and R. F. Freund, “Dynamic mapping of a class of independent tasks onto heterogeneous computing systems,” Journal of parallel and distributed computing, vol. 59, no. 2, pp. 107-131, 1999.
[26] S. Yi, Z. Wang, S. Ma, Z. Che, F. Liang, and Y. Huang, "Combinational backfilling for parallel job scheduling." pp. V2-112-V2-116.
[27] A. W. Mu'alem, and D. G. Feitelson, “Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling,” IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 6, pp. 529-543, 2001.
[28] M. Tahmassebpour, A. M. Otaghvari, “Increase Efficiency Big Data in Intelligent Transportation System with Using IOT Integration Cloud,” Journal of Fundamental and Applied Sciences, vol. 8, no. 3S, pp. 2443-2461, 2016.
[29] M. Wieczorek, A. Hoheisel, and R. Prodan, "Taxonomies of the multi-criteria grid workflow scheduling problem," Grid middleware and services, pp. 237-264: Springer, 2008.
[30] V. Di Martino, and M. Mililotti, "Scheduling in a Grid Computing Environment Using Genetic Algorithms."
[31] K. R. Ku-Mahamud, and H. J. A. Nasir, "Ant colony algorithm for job scheduling in grid computing." pp. 40-45.
[32] J. Kennedy, J. F. Kennedy, R. C. Eberhart, and Y. Shi, Swarm intelligence: Morgan Kaufmann, 2001.
[33] 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.
[34] A. Beloglazov, and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp. 1397-1420, 2012.
[35] C. K. U. K. Kumar, and R. Gogulan, “Max-Min Particle Swarm Optimization Algorithm with Load Balancing for Distributed Task Scheduling on the Grid Environment,” IJCSI International Journal of Computer Science Issues, vol. 9, no. 3, 2012.
[36] J. Cao, and Z. Wu, “An Improved Protocol for Deadlock and Livelock Avoidance Resource Co-allocation in Network Computing,” World Wide Web, vol. 13, no. 3, pp. 373-388, 2010.
[37] M. A. Salehi, B. Javadi, and R. Buyya, “QoS and preemption aware scheduling in federated and virtualized Grid computing environments,” Journal of Parallel and Distributed Computing, vol. 72, no. 2, pp. 231-245, 2012.
[38] M. Arsuaga-Ríos, M. A. Vega-Rodríguez, and F. Prieto-Castrillo, "Multi-objective artificial bee colony for scheduling in grid environments." pp. 1-7.
[39] B. Xu, C. Zhao, E. Hu, and B. Hu, “Job scheduling algorithm based on Berger model in cloud environment,” Advances in Engineering Software, vol. 42, no. 7, pp. 419-425, 2011.
[40] E. Apostol, C. Leordeanu, and V. Cristea, "Policy based resource allocation in cloud systems." pp. 285-288.
[41] F. Lopez-Pires, and B. Baran, “Virtual machine placement literature review,” arXiv preprint arXiv:1506.01509, 2015.
[42] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, Evolutionary algorithms for solving multi-objective problems: Springer Science & Business Media, 2007.