Foundations of Meta-heuristic Algorithms and Cloud Computing

NAIMEH SHARIFI
Department of computer and Information Technology, faculty of Computer and Information Technology, university of Hadaf Institute of Higher Education, iran.
Email: xnax70@gmail.com

Abstract - Cloud computing refers to applications that are delivered as a service over the Internet. The computing model based on high-speed Internet networks is a new model for the delivery and consumption of virtual services. Cloud includes infrastructure, software, platform, and other computing resources using network. One of the problems existing in resource sharing in mobile cloud computing is scheduling the tasks in these cloud networks that are directly related to the issue of efficiency. In order to achieve this goal, scheduling needs to be optimized so that tasks will be implemented on the servers in a balanced way. This makes all servers being optimally used and reduces users' response receiving time. Load balancing is used in mobile cloud networks for this purpose. Load balancing enables tasks to be optimally scheduled on servers and realizes the main goal which is improving resource sharing and increasing network efficiency.

Keywords: Cloud Computing, Meta-heuristic Algorithm, Genetic Algorithm

Introduction

Cloud as a platform offers a variety of services to users regardless of location and hardware in return for payment for service. Cloud computing has received special attention in universities and research centers due to the easier and faster development of web services [1]. Cloud refers to a broad network such as the Internet where an ordinary user is not precisely aware of what is happening behind it. In graphs of computer networks also cloud shape is used to represent the Internet network. The reason for using the Internet infrastructure in the cloud is that it hides technical details of the Internet from the users and creates a layer of abstraction between these technical details and users [2]. "Cloud computing consists of five main features, three service models, and four deployment models" [3].

Cloud computing is completely dependent on the Internet, and all applications and files are on the cloud and on the Internet platform. Cloud computing supports parallel and distributed concepts and shares resources (hardware, software, and information) on demand between users. In return for it, users pay the cost of the resources used. As such, the customer does not need to purchase the operating system or software and can use the needed resources and software at a specified time using the Internet.

The National Institute of Standards and Technology (US-NIST) defines cloud computing as follows: cloud computing is a model for providing easy access based on user's demand over the Internet to a set of changeable computing resources and configuration such as networks, servers, storage space, applications, and services that this category can be provided or released quickly with minimum need for resource management or need for direct intervention by the service provider. Cloud model upgrades accessibility and consists of five essential features of demand-based self-service, widespread network access, resource investment, fast return, and measured service.

Mishra et al. [4] consider load balance and reliability among the most challenging and important issues in a cloud computing environment. The more important challenge is maintaining load balance that can affect memory capacity, load of each processor, and delay or load of the network. Load balance should ensure that all processors at each node of the network have approximately equal loads at any given moment. The above method uses the ant colony algorithm to establish the load balance. The results obtained with parameters of each processor load, response time, network delay compared to the work done have been improved with birds algorithm. Early convergence and increased operational power are among the advantages and getting involved in local optimization is among the disadvantages.

Given the location limitations of the service providing clouds, in most cases, the distance between users and them is high and this causes a reduction in quality of service and increased delay, cost, and energy consumption. Local clouds can be used to address this issue and improve system performance [5]. Limitations of mobile devices, quality of wireless communications, variety of applications, and cloud computing support are all important factors that make the design, programming and deployment of applications on mobile and distributed devices more complex than on fixed cloud devices [6]. Using meta-heuristic algorithms, an optimal way to allocate and share resources in mobile cloud networks can be presented.

Theoretical foundations of research

Cloud network

Cloud computing has been provided as a tool to meet the needs of users [6-7] and users can use its services on the Internet without location dependency. Cloud computing is also used for places where the dynamic provision of resources and the use of virtualization technology are important [7]. Cloud computing is completely dependent on the Internet, and all applications and files are on the cloud and on the Internet platform. Cloud computing supports parallel and distributed concepts and shares resources (hardware, software, and information) on demand between users. Instead, users pay for the resources used.

Therefore, cloud computing in terms of infrastructure refers to a type of distributed and parallel system that comprises a set of interconnected virtual computers. These computers are dynamically provided and are offered as one or more integrated computing resources based on service level agreements [8]. These agreements are established during negotiations between service providers and consumers. Cloud computing seeks to enable dynamic creation of a new generation of data centers by providing services in dynamically networked virtual machines such that application service providers can provide services and applications with greater flexibility and ease and users also can access applications from anywhere in the world.

If mainframes are considered as the first generation of computing systems, we were facing a very large system that users could access through a single terminal. Gradually, these systems became smaller, and by higher processing power, became available to all users as personal computers. Then, it was possible to provide a network with higher processing power through connecting a set of these small systems so that it will meet more and heavier processing needs; but processing needs were increasingly growing and a need for larger and more powerful computing systems was felt [8].

Cloud components

The first component is the users. End users communicate with the customer and manage the relevance and dependency of information in the cloud. Customers are usually divided into three groups [9]. A group of customers are usually smartphones like iPhone and so on. The second group of customers cannot perform computational tasks and only display information and it is the servers that perform the tasks of this group, and the customers have no internal memory. Customers of the third group are mostly Internet browsers which are used to communicate with the cloud; for example Google Chrome, Mozilla, and Internet Explorer.

The second component is the data center of a set of host servers that have different applications. An end user connects to the data center to explain his different needs, and it is also possible that the data center includes many customers from different and far distances. The distributed server is the third component of cloud that can be said that it is the operational power of different applications of Internet hosts; but the use of these applications in cloud is such that the user does not understand it [9].

Cloud computing applications

1. Mobile commerce: Mobile commerce is a model of employment in some fields which is used in activities such as sending messages and information files, financial transactions, shopping, and so on. In this type of application, they face bandwidth constraints, complex device configuration, and security.

2. Mobile training: This application is designed based on e-training and system's mobility. Programs and systems designed in the old method had many limitations such as communication network speed and information storage memory that were resolved by the use of mobile cloud training. Very high storage memories and ultra-fast processors and other cloud computing capabilities increase the efficiency of training through this system.

3. Mobile health care: The suggestion to use this network in medical applications minimized limitations of traditional treatment methods; pharmaceutical errors, limited data storage memories, and security are among them. The mobile health care method makes it easy for users to access resources such as the patient's records. The system also offers medical centers and hospitals to, instead of storing the patients' information on local servers, keep them on cloud systems.

4. Mobile banking: Mobile banking or SMS banking is applicable for some banking and payment activities via mobile phone and tablet. Most of these things are done online today, but they can also be implemented through specific software programs.

5. Mobile gaming and entertainment: This method is used for games that have heavy computations that perform bulky operations via cloud motors and resources, and only displaying it is done through mobile device screens that this reduces power and battery consumption.

Virtualization

Virtualization is the core technology of infrastructure as a serving solution. Virtualization is a way to integrate resources and get more performance from a computer through deceiving it and creating the perception that it is comprised of a set of multiple computers. As Lewis puts it, virtualization is an abstraction of software from hardware. Today, we use this concept to describe when a physical computer is divided into multiple virtual machines and each of these virtual machines runs its own operating systems and software programs independently [10].

Types of virtualization

Virtualization is generally divided into two categories [11]:

- Full virtualization
- Para virtualization

1. Full virtualization

In this case, the machine is fully installed on other machines. The result is that all software programs are placed on the main server.

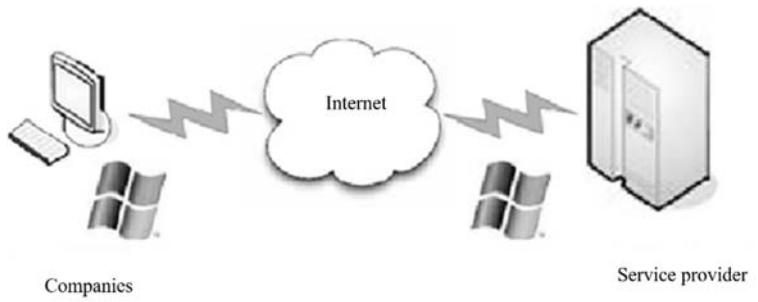

Figure 1: Full virtualization [11]

In this case, the remote data center delivers and manages the services completely virtually. This type of virtualization has been successful in several cases:

- Sharing a computer system between multiple users
- Isolating and separating users from each other and controlling the applications
- Hardware simulation on a virtual machine

2. Para-virtualization

In this case, the hardware can run multiple operating systems on one machine, and this is done by using system resources such as memory and processor.

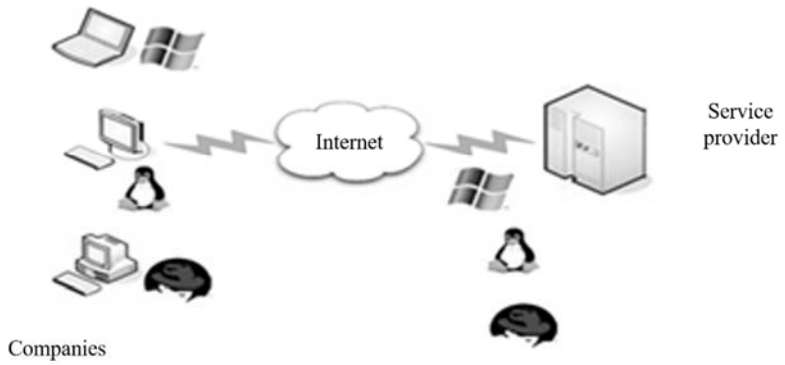

Figure 2: Para-virtualization [11]

Here, services are not fully accessible and in other words, the services are partially provided and made accessible.

Virtual machines

Virtual machines are not different from a physical computer; so, within a virtual machine, any type of operating system can be regulated and any type of service can be created. The main difference between a virtual machine and a physical computer is that physical computer can host a number of virtual machines. This number depends on physical computer resources and the resources we want to allocate to each virtual machine, but on average there can be six virtual machines per physical computer [11].

Virtual machines are a good solution to fully implement software-based virtualization [12]. In fact, using virtual machines in this type of virtualization, hardware with a good quality level is simulated. So, the guest operating system can run on these machines without any changes (Figure 3).

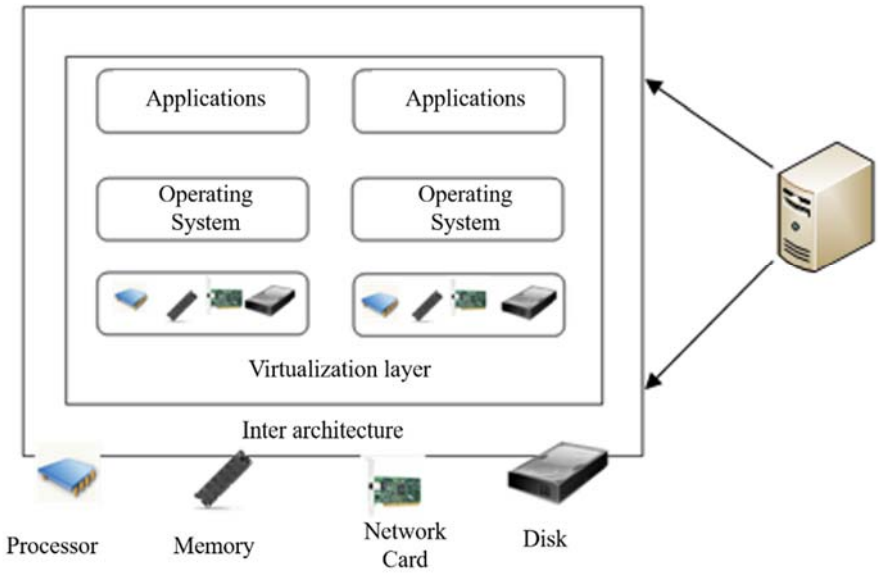

Figure 3: Virtualization [12]

In this environment, two operating systems can simultaneously request access to the same physical resources. Hybrid supervisor controls for physical resources are available between these competing operating system templates.

Load balancing

One of the most important components of a cloud computing architecture is the load balancing component. The most important task of this component is to receive users' tasks and to distribute them on different servers so that access to the data is as localized as possible and the calculations are placed on each server in a balanced way, meaning that the percentage of use of the servers is almost equal.

Load balancing process is, in fact, a general displacement and loading process so that unique nodes in a shared system will have effective use of resources and will improve response times, and it simultaneously divides the tasks between nodes [13]. A dynamic load balancing algorithm does not inherently involve the previous state of the system or behavior of the system and is dependent on the current state and behavior of the system. Important criteria that must be considered in developing this algorithm are [14]:

- Loading estimation
- Loading reason
- Stability of different systems
- Relationship between nodes
- System efficiency
- The essence of the transferred work
- Selection of nodes

Goals of load balancing

Liveny states the goals of load balancing as below [15]:

- Increasing efficiency
- Having a backup plan for a system that has even partially failed
- Providing system stability
- Adjusting the system to reforms

The main goal of load balancing is to speed up the implementation of applications with different resources and to reduce response time [16]. Load balancing methods are used in homogeneous and heterogeneous environments. There are basically two types of load balancing techniques: 1. static, 2. dynamic.

Static algorithms only work when the nodes have low load variation. Therefore, these algorithms are not suitable for cloud environments with high work diversity, and dynamic algorithms are more useful. But to achieve this advantage, more cost must be paid for collecting and maintaining load information.

Dynamic techniques are very successful in balancing tasks across heterogeneous resources. Our proposed method is a dynamic one that, in addition to load balancing, also calculates the priority of tasks in the queue of virtual machines, too.

In cloud environment, whenever a virtual machine has a heavy load of tasks, it can transfer them to the less-loaded virtual machines. In this case, when more than one task is transferred from the busy virtual machine, if there is more than one active virtual machine, these tasks must wait for processing by priority. Load balancing is usually done at the data center level.

Static load balancing: It is examined when compiling the resources needed to maintain load balancing. Running this algorithm is simple but the system overloads. These algorithms can only be useful when the diversity of system tasks is low, so they cannot play an effective role in the cloud environment. Dynamic load balancing algorithms distribute different tasks according to runtime between nodes [23].

Dynamic load balancing algorithm in distributed systems

1. Distributed algorithms

In a distributed system, the dynamic load balancing algorithm is implemented on all current nodes in the system and performs load balancing work as shared among them [17]. The relationship between nodes to achieve load balancing is in two different ways:

- Cooperative
- Non-cooperative

In cooperative mode, the nodes work together to balance the system load, such that for example, all of them aim to reduce response time. In non-cooperative mode, each node works independently to achieve its goal (system load balancing), for example, improving local works response time.

Dynamic load balancing algorithms which are inherently distributed usually generate a lot of messages compared to centralized systems because each node in the system needs to communicate and interact with other nodes; but the advantage of this method is that if one or more nodes in the system fail, there is no reason to stop load balancing algorithm but it affects system efficiency. Distributed dynamic load distribution algorithms can reduce the high pressure resulting from the transmission of information and the status of each node to other nodes in the system. This algorithm is the most useful when most nodes operate individually and have very little relationship with each other.

2. Non-distributed algorithms

In non-distributed mode, any node or a group of them can perform load balancing. Non-distributed dynamic load balancing algorithm is performed in two ways:

Centralized

In this case, the load balancing algorithm runs on only one node in the whole system, called the central node. This node is responsible for load balancing on the entire system. Other nodes are only related to this central node [18].

Partially distributed

In this case, the nodes in the system are divided into clusters that load balance in each cluster is concentrated [18]. In each cluster, a central node is selected that performs appropriate selection techniques to perform load balancing with high precision in each cluster. So, load balancing is applied to the whole system by the central node in each cluster.

In order to come up with suitable resource allocation and scheduling strategy in a cloud, the concepts are inherited from grid computing. The goal of these strategies is to allocate resources such that it can maintain load balancing in the resources and provide service quality standards. Load balancing means that tasks are allocated fairly to the existing resources and over-allocation of tasks to one resource while other resources are free is avoided.

Quality of service means the extent of customer satisfaction, and this criterion is the most important factor in the commercial success of "pay-per-use" systems such as cloud computing. Parameters of the quality of service in cloud computing can be stated as follows:

1. Network bandwidth: When the customer network bandwidth is high, such as multimedia related applications, the cloud must respond to the needs of this bandwidth.

2. Service completion time: For users who have real time demands, scheduling and resource allocation in the cloud should be done in such a way that the service is implemented in the least possible time.

3. System reliability: To run a number of complex user demands, a cloud computing center is required that can provide a reliable efficiency both computationally and in terms of storage services.

4. Cost: One of the criteria considered by users is how much they pay to get a cloud service. If scheduling and resource allocation in this cloud is not done efficiently, cloud costs (including energy-related costs and resource maintenance costs) will increase and this increase in costs will increase user's utilization cost and thus resulting in the reduction of user's satisfaction.

There have been many studies in the area of resource allocation and resource scheduling, both in the cloud computing context and in a grid computing context. These studies and methods can be generally categorized as follows:

1. Traditional scheduling and resource allocation algorithms: These algorithms include round-robin algorithm, weighted round robin algorithm, least relevance scheduling, improved least relevance scheduling, destination hash scheduling, source hash scheduling, etc.

2. Heuristic intelligent algorithms: As mentioned, the issue of scheduling and resource allocation in a cloud and a grid is a full NP issue and thus, the search space of this issue is so large that if an algorithm wants to examine this space sequentially and find the best answer, it will require exponential time. By relying on mechanisms of successful systems, heuristic intelligent algorithms try to simulate these mechanisms to solve difficult problems. Among the heuristic algorithms used in the field of scheduling and resource allocation, ant colony algorithm and genetic algorithm can be mentioned.

Meta-heuristic algorithms

Meta-heuristic algorithms are one of the types of approximate optimization algorithms that have strategies to go out of local optimization and can be used in a wide range of issues [19]. The results of studies by Arian et al. [20] provide a new method for the management of cloud resources that can be effective up to 46% reduction in energy consumption. Efficient resource provision is one of the challenges in the cloud computing environment which has become challenging due to heterogeneous and dynamic resources. Even if virtual machines can handle a large amount of work, software performance is still not guaranteed, and also the diversity of services and quality of service makes it harder to provide sources. Reduced amount of consumed space and saving energy are among the advantages, and delay in doing tasks due to migration is among the disadvantages of this method.

The efficient scheduling algorithm of Ankita et al. [21] showed a 29% reduction in energy consumption and an 8% increase in processing capacity. Reducing energy consumption is a concern of service providers in the cloud computing environment. In line with this, virtual machine stabilization is one of the ways to save energy in cloud data centers. Reducing energy consumption and increasing processing capacity are among the advantages of the above method, and some of its disadvantages include deadline determination and withdrawing the processor from tasks which is time-consuming. Goudarzi et al. [22] presented an algorithm based on dynamic programming in order to create multiple versions of virtual machines without reduction of efficiency and performance in which stabilization of virtual machines in order to minimize energy is done in the data center. One of the advantages of stabilizing virtual machines in a cloud computing environment is the improved reliability of customer service. Nauruki et al. [23] suggest that it may be possible to optimize resource utilization in mobile cloud networks by applying common models used in traditional cloud computing. Modern architectures of mobile cloud networks have outrun single user-single virtual machine usual architecture in terms of reduction in resource consumption. Multi-user architecture- a virtual machine with multiple ownership capability- reduces the amount of resource requirement. Then, the effect of this method is reinforced by the use of multi-user-queue architecture by separating information transfer.

Kao et al. [24] first propose a dynamic programming method for a graph with task-tree structure and divide the problem into sub-problems to reduce the delay. Then, using consecutive trees, the tasks are divided in parallel and eventually, all tasks are merged into one final task. The graph then creates a more complex task, namely, parallel tree chain, that the end path is called for each tree. Mishra et al. [25] have presented different techniques with different parameters. The proposed solution presents Cost Effective Load Balancing Technique (CELBT) which evaluates the parameters of response time, workflow time, and load rate of each virtual machine. The above method is done in three steps. First, the virtual machine is created and two virtual machines for deployment are found and then distributed, and in case of ambiguity in doing the task, the virtual machine is randomly selected. The results show that load balancing is better than other methods. Its advantages include finding the optimal solution in less time and being flexible in finding the solution, and its disadvantages include algorithm sensitivity and having different parameters.

Genetic algorithm

The steps and general structure of genetic algorithm can be described as follows [26]:

1. Start: A random population of chromosomes is generated.
2. Competence: The amount of competence of each chromosome in the population is evaluated.
3. New population: A new population is generated by repeating steps until the new population is complete.
4. Replacement: The newly generated population replaces the previous population to repeat the algorithm another time.
5. Experiment: If the algorithm ending condition is reached, it stops and the best solution is returned from the current population.
6. Loop: Returning to the second step.

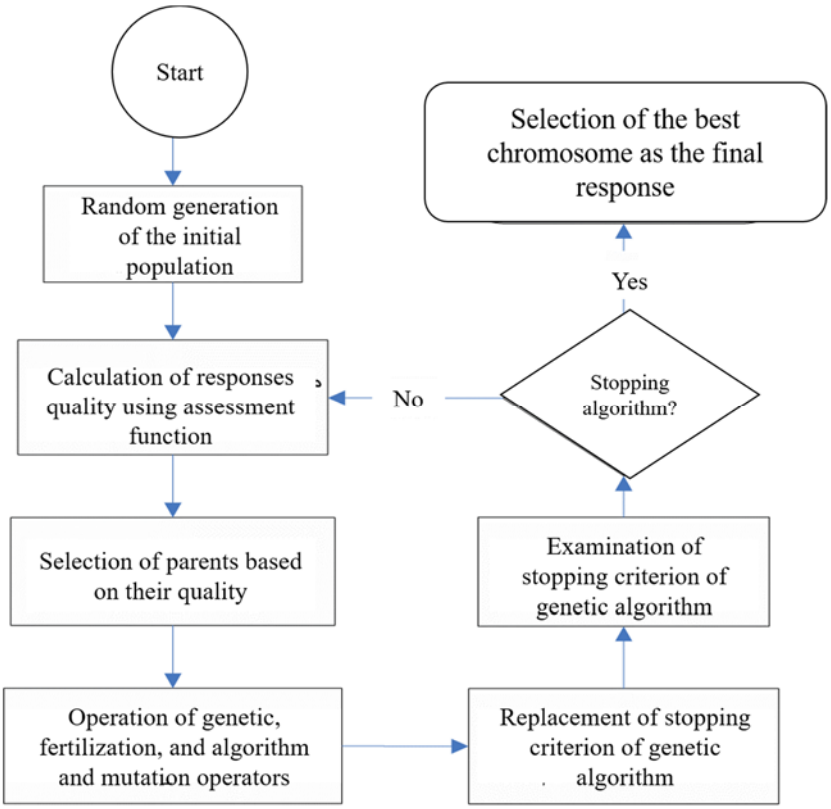

Figure 4: Genetic algorithm general outline [26]

Bee colony algorithm

In Artificial Bee Colony (ABC) algorithm, bees are divided into three groups: employed bees, onlookers and scouts. The bee that remains in the dance area to make the decision to choose a food source is called searcher bee, and the bee that goes toward predetermined food sources is called worker bee. The bee that makes random search is called the leading or scout bee.

In ABC algorithm, for the first time, half of the population of bees is worker bees and the other half is searcher bees. For each food source, there is only one worker bee. In other words, the number of worker bees is equal to the number of food sources around the hive. The worker bees who are tired of working in food sources become scout searcher bees.

In ABC algorithm, each search cycle consists of three stages:

Sending worker bees to food sources and then measuring their nectar value, selecting food sources by searcher bees after sharing information by worker bees and determining the amount of nectar from foods, determining leading bees and then sending them on food sources. At the initial valuing stage, a set of food source positions is randomly selected by the bees and the amount of their nectar is determined. Then, these bees come to the hive and the nectar information of each source is shared with the waiting bees in the dance area inside the hive.

In the second stage, after sharing the information, each worker bee goes to the food source area that it has visited in the previous cycle and that food source exists in its memory; and then a new food source is selected using visual information in the neighborhood of the same one. In the third stage, an onlooker bee chooses the food source area depending on the type of nectar information distributed by the worker bees in the dance area. Somehow, the amount of nectar of the food source increases; and the likelihood of selection of that food source by the onlooker bee also increases. Hence, the dancing worker bees that carry more nectar encourage the onlooker bees to go toward the food source area with more nectar. After entering the selected area, it chooses a new food source in its neighborhood depending on visual information. Visual information is based on a comparison of food source directions. When the nectar of a food source is abandoned by the bees, a new food source is randomly determined by the scout bee and replaces the abandoned source. In this model, in each cycle, a maximum one scout to search a new food source and an equal number of worker bees and onlooker bees go out.

In ABC algorithm, the position of a food source represents a solution to the optimization problem and the nectar value of the food source is associated with the competency of the solution. The number of employed bees or onlooker bees is equal to the number of solutions in the population.

An artificial employed or onlooker bee will likely produce a change in its position (solution) in its memory to find a new food source and test the nectar amount (competency) of the new source (new solution). In the case of the real bee, the production of new food sources is based on the comparison of food sources process in the region related to visual information collected by the bee. In this model, the production of a new food source position is also based on a process of food source position comparison. However, in this model, artificial bees do not use any kind of information in the comparison. They randomly select a food source position and make changes in one of the sources existing in their memory. If the nectar amount of the new source is greater than the previous one kept in the bee's memory, it maintains the new location and forgets the previous one. Otherwise, it maintains the previous position. After the search process of all worker bees is completed, they share nectar information of food sources (solution) and the information related to their position with the onlooker bees in the dance area. An onlooker bee evaluates nectar information taken from all working bees, and a food source with probability related to its nectar value is selected. Also, in the case of the employed bee, it examines the generation of changes in the position (solution) existing in its memory and the nectar value of the selected source (solution). It offers the nectar that is greater than the previous one. The bee keeps the new position and forgets the previous one. The onlooker bee selects a food source according to the probability value related to that food source, *pi*, which is calculated as below:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad (1)$$

Where fit_i is the competence value of solution *i* evaluated by its employed bee, that the evaluation is proportional to the value of the nectar existing in the food source at position *i,* and SN is the number of food sources which is equal to the number of employed bees (BN). In this method, employed bees exchange their information with onlooker bees. In order to produce a selected food source from the previous one, ABC uses the following statement:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \qquad (2)$$

Where the indices of $k \in \{1, 2, \ldots, BN\}$ and $j \in \{1, 2, \ldots, D\}$ has been randomly selected. Although K has been randomly determined, it is different from *i*. $\varphi i,j$ is a random number between [1 and -1]. It controls the production of the neighboring food source position around xi,j and the comparative changes of the neighboring food positions is provided by the bee visually. Equation 2-3 shows various parameters between xi,j and xk,j ; also the changes in the position xi,j are reduced. Therefore, somehow the search for the optimal solution in the search space is approached and the step decreases consecutively. If the parameters generated by this operation are more than their predetermined value, the parameter can be selected as an acceptable value. The food source that its nectar is abandoned by the bees is replaced by a new food source by the scout bees. In ABC algorithm, this is simulated by randomly generating the position and replaced the abandoned source. In ABC algorithm, if a situation does not improve more than a predetermined number of cycles called the limit, then the food source is

assumed to be abandoned. After selecting each source, the *vi*,*j* position is produced and then evaluated by the artificial bee; then, its performance is compared with *xi*,*j*; if the new food sources have equal or better nectar than the previous source, it will replace the previous one in memory; otherwise, it keeps the previous one. In other words, a greedy selection mechanism performs the choice between previous and current food sources.

In the case of bees, agents are used to measure how quickly the bee finds the colony and exploits the newly discovered food source. Artificial employment can similarly show the speed at which the solution is possible or discover solutions with high quality for complex optimization problems. Survival and progress of the bee colony depend on the rapid discovery and efficient use of the best food sources. Similarly, the right solution to difficult engineering problems is related to the quick discovery of specific good solutions to problems that need to be resolved in the real time. In a robust search process, exploration and exploitation processes must be done together. In ABC algorithm, while employed and onlooker bees do the exploitation process in search space, the scout bees control the discovery process.

Conclusion

Bee algorithm has two major advantages over other algorithms: automatic segmentation and the ability to deal with multi-quality problems. The bee algorithm is based on congestion intelligence. This issue results in an automatic division of the entire population into subgroups with a given mean distance, and each group can congregate around a local optimal. Among all these optimums, the best global optimum can be found. Secondly, this subdivision allows for optimization of all, especially for nonlinear multi-quality optimization problems. For bee algorithm, random control is set relative to the repetition, such that convergence can also be accelerated by tuning these parameters. These benefits make dealing with issues of consistency, clustering, classification and also hybrid optimization suitable.

In the proposed method, the speed to obtain the response significantly increases and at the same time, response accuracy is also much higher. The advantage of the bee optimization algorithm is that it converges quickly, but near the optimum point the search process slows down sharply. On the other hand, the bee algorithm is highly sensitive to initial conditions. In fact, the random nature of bee algorithm operators makes the algorithm sensitive to the initial population. This dependence on initial conditions is such that if the initial population is not selected well, the algorithm may fail to converge or not yield a good response. For this purpose, in order to obtain a better answer, in the proposed method, the random method is used to value the initial population in order to create the best initial population.

Bee algorithm has important advantages compared to standard optimization methods:

1. Parallel processing is one of the most important advantages of the bee algorithm. This means that in this method, instead of one variable, a population is grown toward an optimal point at a time. So, the convergence speed of the method increases very much.

2. Random search nature of this algorithm in problem space is somehow considered a parallel search because each random bee generated by the algorithm is considered a new starting point for searching part of the problem state space and search in all of them occurs simultaneously.

3. Due to the breadth and dispersion of the search points, good results are obtained in problems that have large search space.

4. It is considered a kind of targeted random search and will achieve different answers from different paths. In addition, it faces no restrictions on the path of searching and selection of random answers.

References

[1] L. M.Vaquero, L.Rodero-Merino, J. Caceres and M.Lindner.“A break in the clouds:Towards a cloud definition”, Sigcomm Computer Communications Review, Vol . 39, No . 1, pp : 30-35, 2009 .
[2] Rajat banerji.“ Elastic Load Balancing in Amazon Compute Cloud”. harward university-march. , 2011.
[3] J.Broberg, S.Venugopal and R.Buyya,“Market-oriented Grid and utility computing:The state-of-the-art and future directions”, Journal of Grid Computing Elsevier, Vol . 6, No . 3, pp : 255-276, 2016.
[4] Gundipika Kaur, Kiranbir Kaur,“An Adaptive Firefly Algorithm for Load Balancing in Cloud Computing”, Springer Proceedings of Sixth International Conference on Soft Computing for Problem Solving, pp:63-72, 2017.
[5] WoodT.“Black-box and Gray-box Strategies for Virtual Machine Migration”, Proceedings of the 4th Int’l Conference on Networked Systems Design & Implementation, IEEE. , 2017.
[6] B.Yagoubi,Y.Slimani,“Task load balancing strategy for grid computing”, Journal of Computer Science, pp:186-194 , 2015.
[7] A.Revar, M.Andhariya, D.Sutariya and M.Bhavsar,“Load balancing in grid environment using machine learning-innovative approach”, International Journal of Computer Applications, pp :88-98, 2010.
[8] M.Randles, A.Taleb-Bendiab and D.Lamb ,“Scalable self governance using service communities as ambients”, Proceedings of the IEEE Workshop on Software and Services Maintenance and Management (SSMM) within the 4th IEEE Congress on Services, IEEE Services-i, 2009.
[9] L.M.Vaquero, L.Rodero-Merino, J.Caceres, and M. Lindner, “A break in the clouds:Towards a cloud definition”, Sigcomm Computer Communications Review, Vol . 39, No . 1, pp : 50-55, 2016.
[10] Rakesh Kumar., Neha Gupta., Shilpi Charu, Kanishk Jain and Sunil umar Jangir ,“Open Source Solution for Cloud Computing Platform Using OpenStack” , International Journal of Computer Science and Mobile Computing , Ijcsmc, 3(5), pp:89-98 , 2015.
[11] Shridhar.G.Donamal,"Optimal Load-Balancing in Cloud Computing by efficient utilization of virtual resources", IEEE, 2014.

[12] V.Venkatesa Kumar.,R.Revathi and Newlin Rajkumar ,"An Assessment on Various Laod Balancing Techniques In Cloud Computing" ,IJAICT, 1(8), 2014.
[13] Jasmin James and Dr. Bhupendra Verma,"Efficient VM load balancing algorithm for a cloud computing environment", pp:1258,1368, 2016.
[14] Mamta Khanchi and Sanjay Tyagi,"An Efficient Algorithm for Load Blancing In Cloud Computing", International Journal of Engineering Sciences & Research Technology , IJESRT, 2016.
[15] Dhinesh Babu L.D and P.Venkata Krishna,"Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied Soft Computing ,Vol.13, No.5 ,pp: 2292-2303, 2013.
[16] M.Livny and M.Melman, "Load Balancing in Homogeneous Broadcast Distributed Systems", ACM Comput. Network Performance Symp., Vol . 11, No . 1, pp : 47-551982.
[17] D.L.Eager., E.D.Lazowska and J.Zahorjan ,"Adaptive load sharing in homogeneous distributed systems", The IEEE Transactions on Software Engineering, Vol . 12, No . 5, pp : 662-675, 1986.
[18] R. Mirchandaney., D. Towsley and J. Stankovic ,"Adaptive Load Sharing in Heterogeneous Distributed Systems," Journal of Parallel and Distributed Computing,Vol.9 ,No.4, pp: 331-346 , 1990.
[19] Fang Liu., Jin Tong., Jian Mao., Robert Bohn., John Messina., Lee Badger and Dawn Leaf ,"Nist Cloud Computing Reference Architecture", Elsivier. 2011.
[20] J. Broberg., S. Venugopal, and R. Buyya,"Market-oriented Grid and utility computing: The state-of-the-art and future directions", Journal of Grid Computing, Vol. 6, No. 3, pp: 255–276, 2008.
[21] Ehsan Arianyan., Hassan Taheri and Saeed Sharifian, "Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data", Computers & Electrical Engineering,Vol.47, pp:222-240 , 2015.
[22] Ankita Choudharya., Shilpa Ranab and K.J. Matahaic, "A Critical Analysis of Energy Efficient Virtual Machine Placement Techniques and its Optimization in a Cloud Computing Environment", International Conference on Information Security & Privacy (ICISP), IEEE, pp:11-22, 2015.
[23] Hadi Goudarzi and Massoud Pedram,"Achieving Energy Efficiency in Datacenters by Virtual Machine Sizing, Replication, and Placement",Vol.100 , pp:161-200 , 2016.
[24] Dhinesh Babu L.D and P. Venkata Krishna,"Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied Soft Computing ,Vol.13, No.5, pp: 2292-2303, 2013.
[25] Kao, Y.H., et al,"Hermes: Latency optimal task assignment for resource-constrained mobile computing", Vol.16, No.11 ,2017.
[26] Ratan Mishra and Anant Jaiswal,"Ant colony Optimization: A Solution of Load balancing in Cloud", International Journal of Web & Semantic Technology (IJWesT), Vol. 3 , No.2 ,pp:3-20, 2012.
[27] Songnian Zhou ,"A Trace-Driven Simulation Study of Dynamic Load Balancing" , IEEE Transaction On Software Enginearing, pp:14-9 , 1988.