Science Arena Publications
Specialty Journal of Electronic and Computer Sciences
ISSN: 2412-7485
Available online at www.sciarena.com
2020, Vol, 6 (1): 1-12

Presenting a Method for Optimal Service Selection in Cloud Computing Environment Using Genetic Algorithm

Masoud Karimi[1], Vahid Saffari[2]*

[1]Department of Computer Engineering, Faculty of Engineering, Islamic Azad University, Shiraz, Iran.
[2]Faculty member of Islamic Azad university of Iran, Shiraz Branch, Shiraz, Iran.

***Corresponding Author**

Abstract: *Cloud Computing (CloCom) is one of the latest technologies to speed up and improve the quality of service delivery as well as system services being based on communication networks, including the Internet. An important point in this regard is that in CloCom, there is no need for the service recipients' awareness of infrastructure and specialized information provided by cloud service providers. This is the reason for using the term cloud. There are many challenges in terms of using CloCom, and one of the most important challenges concerns the selection of optimal service (C_SS) in this environment. Due to different conditions of CloCom's service providers, such as VM Price / hour and VM CPU_Virtual_Core, it is necessary to allocate more efficient C-SS to these service providers. For this purpose, the genetic optimization (GA) method for C_SS has been used. To this end, the genetic optimization (GA) method has been used for C_SS. Each service provider can be assigned a service based on the ability to perform cloud requests (C_SReq) using the design of the GA optimization method for C_SS among CloCom servers. The evaluation results indicated a decrease in average migration time and average response time in CloCom, which eventually has led to a higher C_SReq and an increase in the quality of service. The average decrease in response time and migration time in the C_GSS method was extracted and compared with randomized C_SS (C_RSS) and particle mass (C_PSS). The results indicated the reduction of the mentioned criteria C_GSS research methodology due to C_SS optimization in CloCom service providers.*

Keywords: *Cloud computing, Service selection, GA model, Migration time, Quality of service.*

INTRODUCTION

CloCom is a form of payment, whose using enables the availability of a suitable demand-based network to a host of common computing resources such as networks, servers, storage of applications and services. These resources can be quickly provided and sent with the minimum managerial effort and interactions with the service providers. This cloud-based approach improves availability and includes five key features: on demand self-service, network access everywhere, location independent resources, quick flexibility, and payment by use. (Birje et al., 2017) CloCom has several benefits, including fewer system costs, expanded performance, lower software costs, fast and permanent software upgrades, unlimited storage capacity and easier team collaboration. (Begum and Prashanth, 2013)

In general, the cloud computing architecture consists of three layers: software as a service, a platform as a service, and an infrastructure as a service. (Birje et al., 2017) Considering the emergence of a variety of cloud service providers, the selection of the most appropriate service for cloud computing has been of great importance, and one of the important issues in this field is the comparison and the optimal C_SS. The cloud service broker, being cloud based, is an interface between cloud providers and clients helping users compare services and offer the most appropriate service. (Qu, Wang andOrgun, 2013)

Cloud service brokers are software agents that provide final access points to web servers. (Chen and Mohapatra, 2005) In fact, Cloud Service broker is a cloud-based interface between the cloud provider and consumer, helping cloud clients choose the best service and offer the most appropriate service. (Quarati et al., 2013) The cloud-based broker of cloud acts as an intermediary between the buyer of a CloCom service and the provider of that service and saves time. (Sundareswaran, Squicciarini and Lin, 2012)

Different analyses for appropriate C_SS can be done by human factors such as experts or software agents such as service-monitoring services that analyze the parameters of the quality of the services of the candidate. This variation provokes the C_SS, especially when there is a wide range of providers and a large number of features for the service criteria. (Sharifi et al., 2014)

The most important criteria for measuring the performance of cloud services include the CloCom response (Singh, 2014), sustainability (Danish and Sharma, 2018), competence (Angabini et al., 2011), accuracy (Danish and Sharma, 2018), transparency (Danish and Sharma, 2018), establishing collaboration capability (Danish and Sharma, 2018), availability (Sabahi, 2011), reliability (Sabahi, 2011), stability (Kiruthika and Khaddaj, 2013), cost (Singh and Jangwal, 2012), compatibility (Dastjerdi and Buyya, 2014), elasticity (Coutinho et al., 2015), usability (Sabahi, 2011), efficiency (Sabahi, 2011), and service scalability (Ab Rashid Dar, 2016).

Many methods have already been proposed for the optimal C_SS problem in the CloCom environment, the most recent of which is the use of optimization algorithms and expert systems. To design and set up an intelligent and expert system using efficient optimization algorithms for optimal C_SS in the CloCom environment, several issues must be considered. Several optimization algorithms have been proposed to improve the optimal C_SS in the CloCom environment, but in most of the proposed algorithms, due to the insufficiency of parameters considered to optimize the services in these algorithms (one or two parameters), the migration time of service is increased and the quality of service in the cloud computing environment is decreased. With the knowledge of service quality, C_SS is a key requirement in service-based computing because it enables complex cloud client operations to meet the quality of service constraints (Lan et al., 2009). Quality-based C_SS was first reported by Chang et al, and was then considered by many scholars (Zhang et al., 2010).

In 2007, Huffman introduced a planning equivalence methodology to illustrate the combination of Web services and the identification of a particular case of Web service combination called "forward-looking" (Hoffmann, Bertoli and Pistore, 2007). In 2002, Grossman used non-linear integer programming techniques as an alternative for using the GA model to find the optimal combination of services (Grossmann, 2002). In 2010, Zhang et al. presented an ant clony optimization technique for choosing a dynamic combination of Web services based on service quality (Zhang et al., 2010). Cau et al. proposed the C_SS model, taking into account time, quality, cost, and service criteria (Cao et al., 2016). Xiang and colleagues analyzed the service mixing problems in a wide range of data (Xiang et al., 2016).

The GA model was presented by John Holand in the University of Michigan in 1975 (John, 1992). In 1992, John Kouza used the GA model to resolve and optimize advanced engineering issues and was able to put the GA model into the computer language for the first time. The most important components of the GA models are chromosome, population, and fitness function (Mukhopadhyay et al., 2009). The GA model is inspired by the concepts of reproduction in nature using evolutionary rules. In this algorithm, search agents are a set of chromosomes and this method is used to optimize various issues using variables such as selection, mutation, and promotion, being based on the fit function defined for each problem.

The research problem can be called the C_SS problem solving; the solution method presented in this study is based on the GA model; C_SS between the CloCom servers uses the properties of each CloCom server. In this research, the first part has reviewed the research literature and articles published in relation to the subject, the second part is devoted to the research methodology, and the third part of the research has discussed the findings.

Research Physiology

In this research, using the GA model, a C_SS approach has been presented that, by choosing the optimal service for each CloCom service, ultimately reduces the migration time of the CloCom and improves the quality of service in this environment.

C_GSS Research Method

At first and after configuring the components of each system in the environment intending to conduct C_SReq, the requests are sent to the Cloud broker, the Cloud broker continues to examine the CloCom Service Evaluation Unit by choosing the best host as well as virtual machine performing this C_SReq and provides it to the cloud-based broker. Cloud broker sends C_SReq to the virtual machine introduced to perform this C_SReq, receives the result of C_SReq, and delivers them to the requesting system.

As shown in Figure 1, in the C_GSS method, at first and at the moment of the start of the C_SReq, the CloCom server delivers values of the VM Price / hour and VM CPU_Virtual_Core properties as inputs to the C_GSS survey. The GA model of research, after receiving the terms of each CloCom server at any given time, using the optimized variables of the C_GSS research method, decides to determine the optimal CloCom server based on the ability and capacity of each CloCom server regarding the recieved inputs.

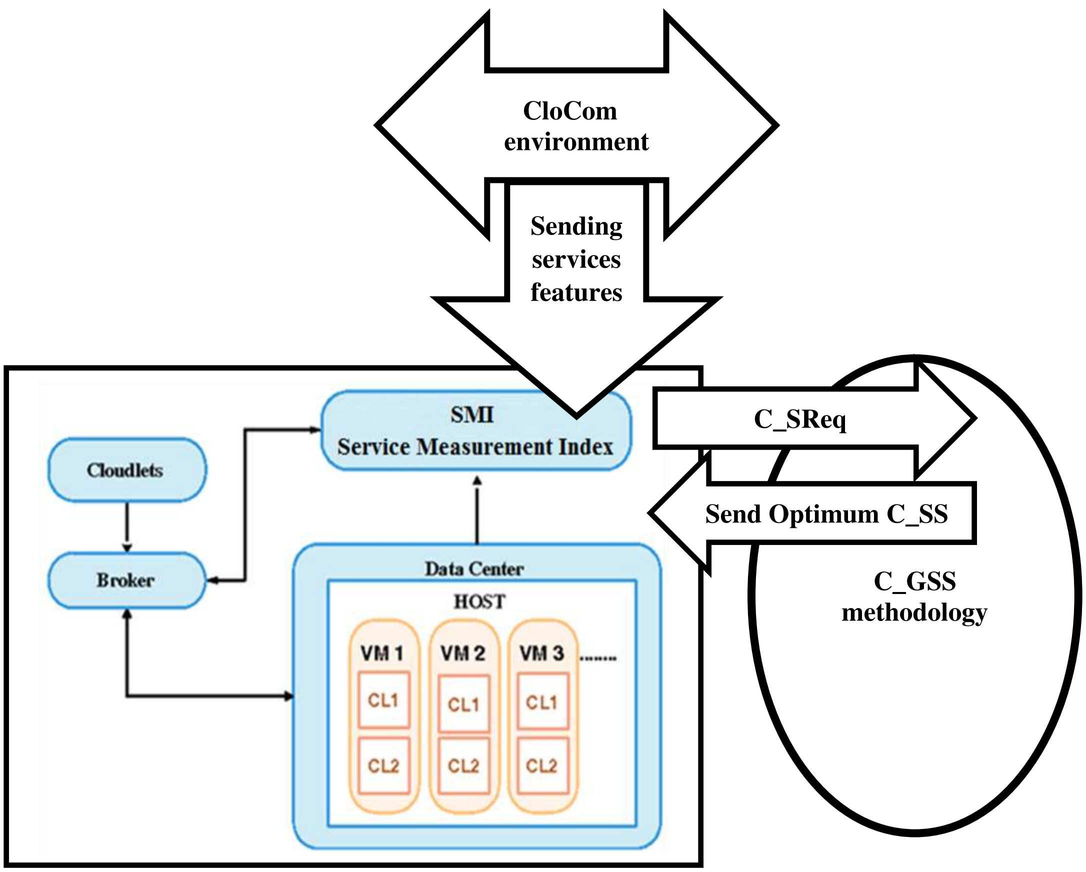

Figure 1: The procedure for performing C_SReq by different units in CloCom

The operation of C_GSS method to obtain C_SS optimization for CloCom is presented in figure (2) in a flowchart format.

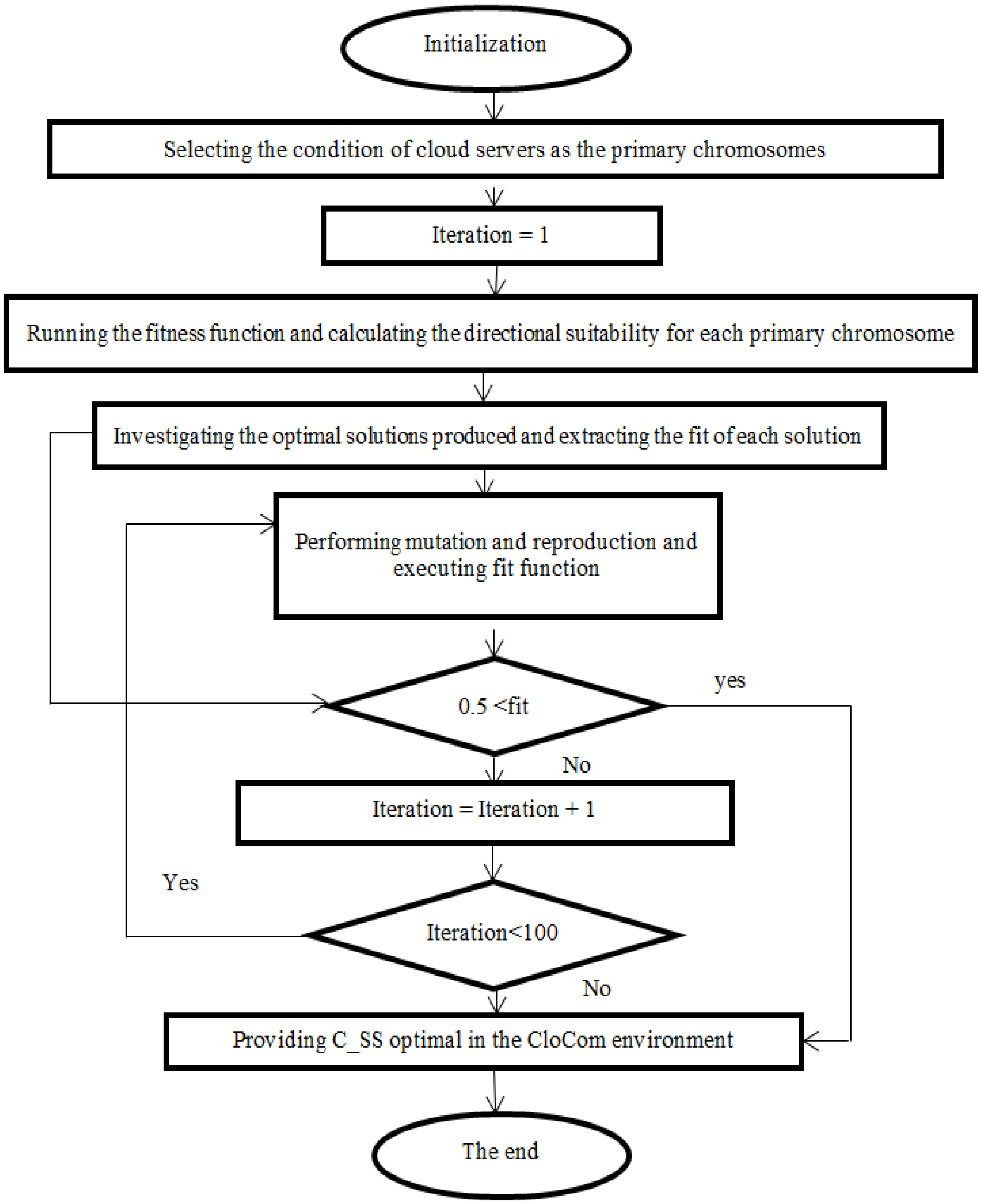

Figure 2: Flowchart of the function of the C_GSS research method

In C_GSS, several variables should be designed and determined in a variety of ways. Given that the research problem has been implemented and evaluated in the context of a simulated CloCom environment, the design and designation of the variables of the C_GSS research method is a test and error method calculating the extent that the desired results are obtained by the C_GSS research. In this regard, the C_GSS variables are changed. The optimal values of C_GSS variables, which are calculated by the test and error method, are presented in Table (1).

Table 1: Optimal values of designing variables in C_GSS research method

variables	**Status / amount**
Number of optimization variables	2
The number of optimization cycles	100
Population size	80
The minimum cost	0,5
Type of selection function	Roulette wheel function
Type of Mutation function	Uniform
Mutation Rate	0.9
Type of Crossover function	Single point
Crossover Rate	0.1
Constraints of research optimization	=< 100VM CPU_Virtual_Core0 < =<100VM Price/hour0 <
Expense function	)VM Price/hour) / (VM CPU_Virtual_Core(

In each run of the C_GSS implementation, which is designed using the variables presented in Table (1) and simulated in the MATLAB programming language, the CloCom's path and service points is estimated by the cost function being based on the constraints listed in Table (1). After calculating the score of each CloCom server, each CloCom server is selected optimally in each cycle of the CloCom server. In the last cycle, the optimal CloCom, which has the lowest distance to the C_SReq demanding system, as well as the optimal conditions in terms of VM Price / hour and VM CPU_Virtual_Core as the destination of C_SReq being announced by the CloCom service are declared to the cloud service broker.

In order to be able to evaluate the C_GSS research method, it is necessary to simulate this methodology by a suitable programming environment. In this research the mathematical programming environment R2017b has been used, and in the following sections, simulations of the C_GSS method of research in the programming language are described.

Before starting to simulate the C_GSS method, the research of various concepts and components of CloCom, including the CloCom service measurement, Cloud broker, virtual machines, and host systems, are designed in this simulation.

Simulation of the C_GSS Method in Matlab

Considering the optimal extracted variables for designing the C_GSS method of research, being offered in Table (1), this parameter has been encoded in the MATLAB programming environment and implemented and coded in the form of M various files of cost, selection, and mutation functions. In addition, due to the fact that the C_GSS research method is compared with two (Yang et al., 2014) C_PSS and C_RSS, these two methods are also simulated in the MATLAB programming language in the simulated CloCom context in this programming environment.

The simulation method of the C_GSS is that in the beginning, C_SReqs are produced by the cloud service providers with different features, and these requests are sent to the cloud broker and then to the CloCom service measurement to select the optimal C_SS. The CloCom Service Measurement Unit selects the best CloCom server to run C_SReq in all three methods simulated in MATLAB and in the simulated CloCom platform. After sending the received C_SReq cloud requests to the selected server in all three methods, the simulation is eventually completed by issuing its successful execution message C_SReq in the C_GSS research procedure in accordance with the following procedure if that CloCom server has the ability to run C_Sreq.

Then the same C_SReq is executed in C_RSS and the CloCom client which was selected for C_SS is also chosen in this method and C_SReq can be sent to them. All procedures mentioned above, from the beginning of the entry of C_SReq to the successful or failed execution of these requests, are ultimately enforced by the C_PSS method (Yang et al., 2014).
At each step that C_SReq is simulated to CloCom and the C_SS is adopted by the CloCom service measurement unit, it is optimized in different periods of C_GSS and (Yang et al., 2014) C_PSS as well as optimal C_SS for executing requests. An important point in the C_GSS research, C_RSS, and C_PSS CloCom method is that the better are the server choices, the more successful C_SReqs will be added and CloCom's migration time will be reduced. After comparing the C_SReq 200 with different conditions in C_GSS research and C_SS, the criteria for measuring the service accuracy, error of execution, the average migration time, and the average response time can be obtained according to the following equations.

Equation (1)

$$\text{Cloud service accuracy} = \frac{\text{the number of successful implementation service}}{\text{the total number of cloud sevices}}$$

Equation (2)

$$\text{Cloud serviceerror} = \frac{\text{the number of unsuccessful implementation service}}{\text{the total number of cloud sevices}}$$

Equation (3)

$$The\ average\ of\ migration\ time = \sum_{i=1}^{n} \frac{\text{i migration time in cloud service}}{\text{(n)the total number of cloud sevices}}$$

Equation (4)

$$The\ average\ of\ response\ \ time = \sum_{i=1}^{n} \frac{\text{i response time in cloud service}}{\text{(n)the total number of cloud sevices}}$$

In Table (2), all CloCom hypotheses simulated in MATLAB are presented as the basis for evaluating comparative methods of this research.

Table 2: CloCom simulations hypothesized in MATLAB programming language

Number of CloCom data centers	Number of CloCom host service providers	The number of CloCom virtual machines	CloCom's default idle time
4	5	15	15 time units

In the following, a series of C_Sreqs has to be defined for conducting in this CloCom environment; the properties of each C_SReq in this simulated environment are also in accordance with Table (3).

Table 3: C_SReq properties in the CloCom simulated in MATLAB

Number of C_SReq	C_SReq creation method	C_SReq features
200	random	Number of processor cores required (range 0 to 100%) And runtime

Findings

After simulating CloCom in the MATLAB programming language, the service accuracy is presented in Figure 3, C_GSS research, C_RSS and (Yang et al., 2014) C_PSS in all three methods.

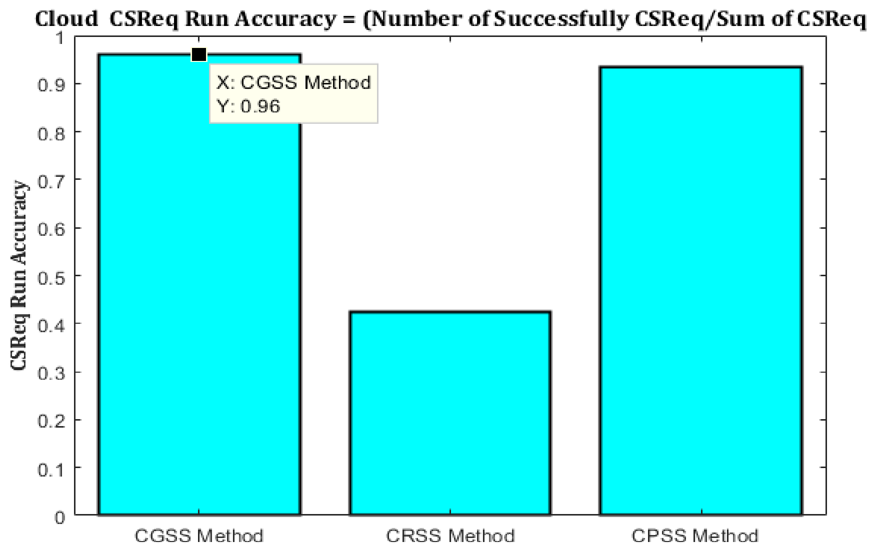

Figure 3: Comparison of service quality criteria in comparative methods

As shown in Fig. 3, the accuracy of the service is calculated in the C_GSS research method, which is 0.535% higher than the C_RSS method and 0.025% higher than C_PSS. A greater success in the criterion of service accuracy in the C_GSS method of research suggests that C_SS has been done more precisely to perform C_SReq in the C_GSS research approach.

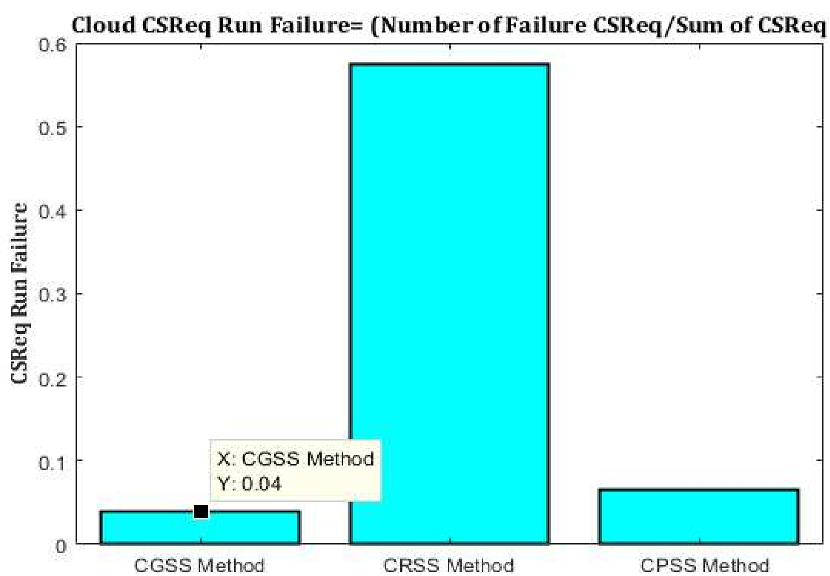

Figure 4: Comparison of Service Error Criteria

As shown in Figure 4, the service error rate in the C_GSS survey was less than 0.535% of the C_RSS and was less than 0.025% less than (Yang et al., 2014) C_PSS, which have been evaluated in the following. The survey would eventually reduce the migration time in the simulated CloCom.

Considering the fact that each C_SReq can not be run by the CloCom server, an amount of time has been spent as the migration time being re-sent to the cloud broker unit and queried from the CloCom's Client Service Unit to the new host server.

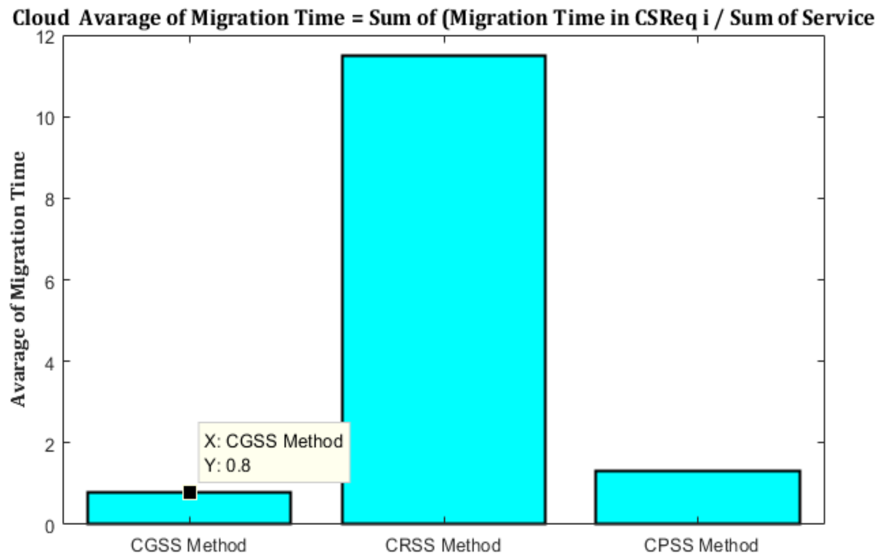

Figure 5: Comparison of Average Migration Time of CloCom

As shown in Fig. 5, the average migration time in the C_GSS research was 0.8 times unit, which was 7.10 time unit less than C_RSS and 0.5 time unit less than (Yang et al., 2014) C_PSS. Thus, with the optimal selection of the CloCom service provider, the migration time of the server can be significantly reduced, and CloCom's migration time would ultimately reduce the response time and service quality of the environment.

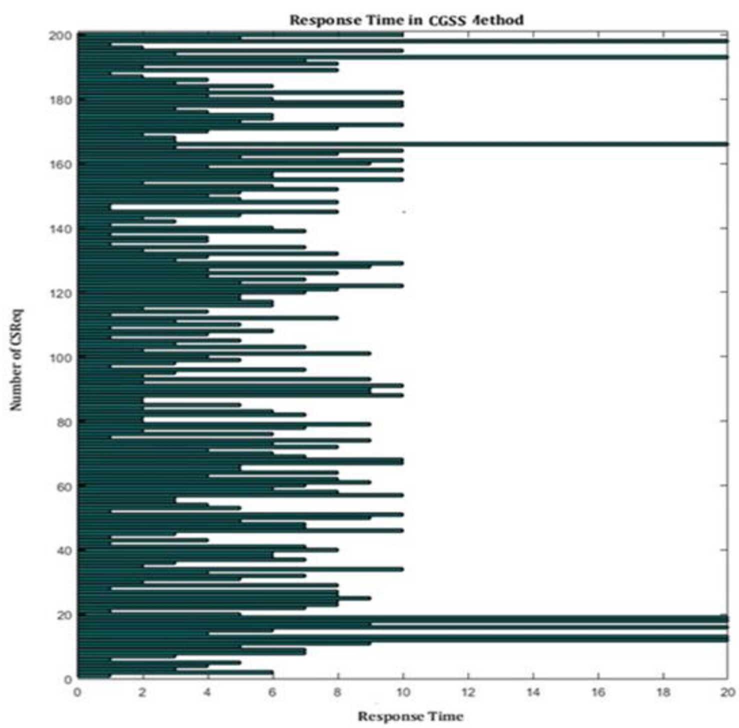

Figure 6: The investigation of response time for 200 C_SReq in C_GSS research

As shown in Fig. 6, in C_SReq 200, 10 time units, has been spent as the response time in the C_GSS research method, all of these runtimes can be observed in each of the C_SReqs 200 in the CloCom environment in Fig. 6.

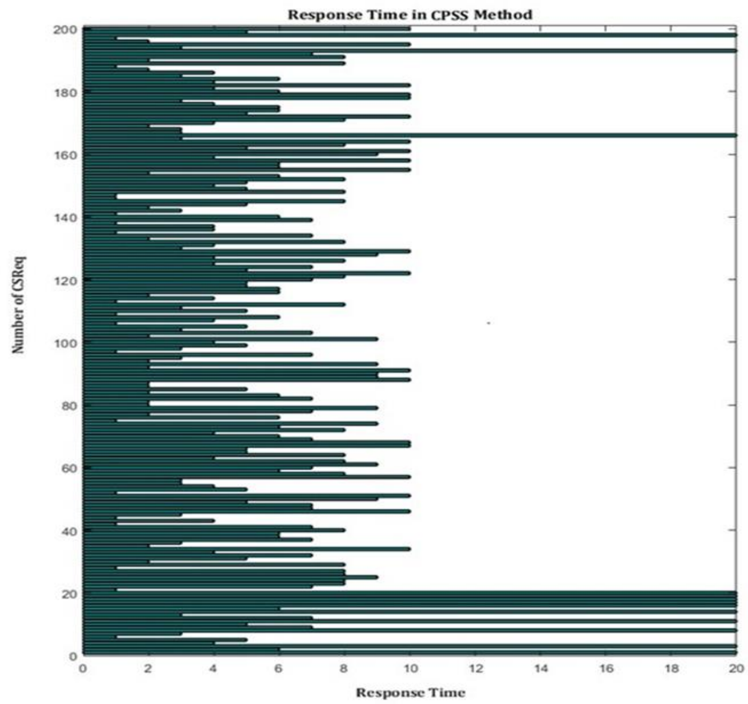

Figure7. The investigation of response time for C_SReq200

For example, the response time in C_PSS for C_SReq 200 was 10 times. This was equal to the response time of the same C_SReq in the C_GSS research procedure. In the following, all the steps mentioned for C_RSS were also implemented and the results of the CloCom response time in this method for the C_SReq 200 obtained in Fig. 8 have been provided.

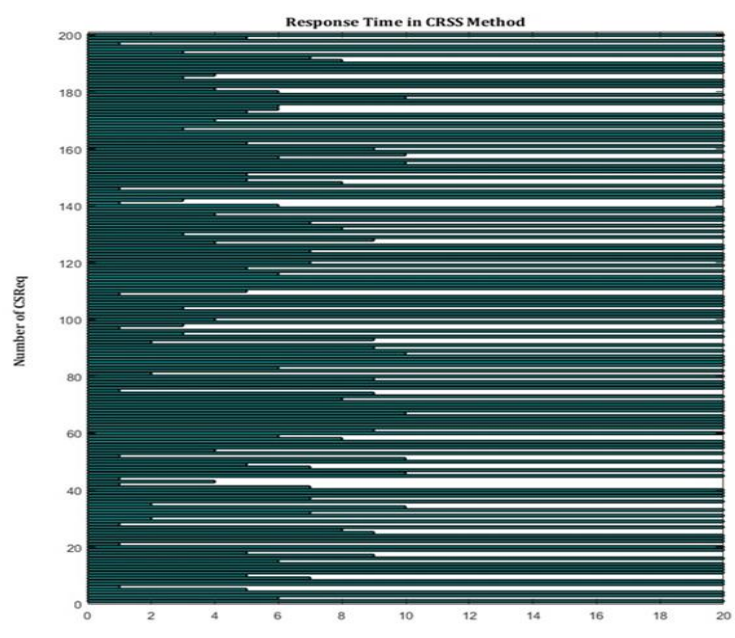

Figure 8: The investigation of response time for C_SReq 200 in C_RSS

It can be seen in Fig. 8 that in C_RSS 20 time units has been spent for C_SReq 200, the extracted value for this C_SReq was 10 times greater than the suggested C_SS method.

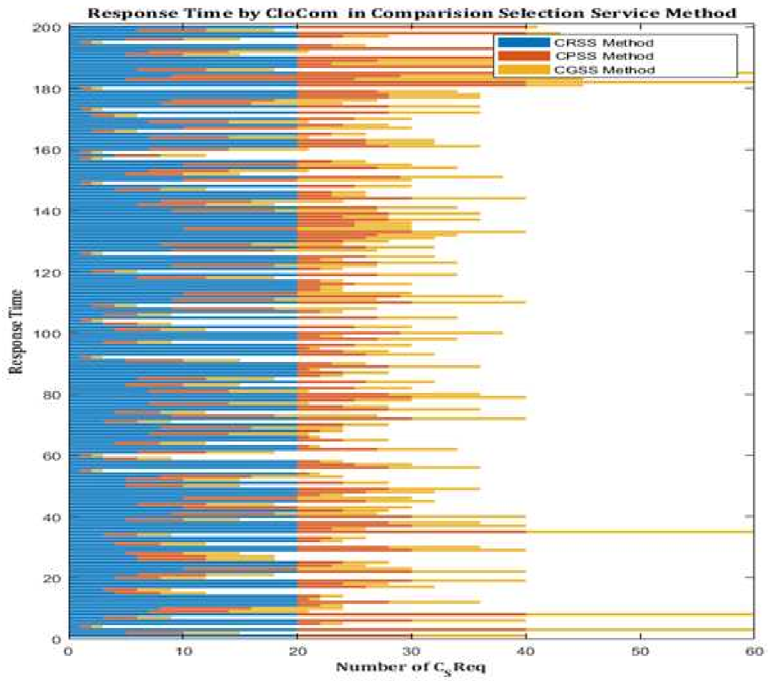

Figure 9. The investigation of the response time for C_SReq 200 in the research methods

As seen in Figure 9, the majority of C_SReq executed in CloCom, the response time in the C_RSS and (Yang et al., 2014) C_PSS methods were greater than those of the C_GSS method. In the following section, the findings of the evaluation of this research have been summarized and analyzed.

Conclusion

The main purpose of this study was to provide a method for reducing the migration time of CloCom, which ultimately would lead to an increase in the quality of service in CloCom. One of the most important criteria for measuring the quality of services in CloCom, which leads to the comprehensive and operational use of this environment in the future, is the response time criterion. The lower the response time in CloCom, the more C_SReq will be achieved in this environment, and cloud vendors will get faster results, and in fact, the quality of the service will be enhanced in this computing environment. The average response time criterion in the C_GSS method of the research was 01.6 times, which was 7.8 times less than the C_RSS and 0.36 times less than (Yang et al., 2014) C_PSS. This less than average response time in the C_GSS research method has been obtained by the comparative methods due to the fact that in the C_GSS study, in addition to considering the conditions of each CloCom service providers, routing was conducted, and the CloCom service providers that have the best conditions and the minimum distance from the C_SReq system have been chosen, which will would result in a lower amount of CloCom migration time and response time in the C_GSS research approach than the comparator methods.

Reducing CloCom's migration time and response time would further enhance the quality of service in the CloCom environment.

According to the results, the most important reasons for reducing the migration time of CloCom and the response time that has enhanced the quality of service in CloCom have been described in the C_GSS research method when compared to the comparative methods as follows:

1. The GA model achieved a good result in issues of great search space, such as CloCom environments, because of the breadth and dispersion of the points that are being searched.
2. The GA model was considered to be a randomized target search, and came up with different solutions from different paths. In addition, it was not confronted by any limits in random search on the path, which is very suitable for optimal selection due to the dispersion of cloud service providers in CloCom environments and the variable features of these servers for optimal selection.
3. Due to the competition, the answers and the selection of the best of the population would be achieved with high probability and precision at the optimal level in the research question (optimal C_SS).
4. Its implementation was simple and did not require complicated problem-solving procedures, so it had an appropriate speed for solving the problem of this research.

References

1. Ab Rashid Dar, D. D. R. (2016). Survey on Scalability in Cloud Environment. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *5*(7).
2. Angabini, A., Yazdani, N., Mundt, T., & Hassani, F. (2011, October). Suitability of cloud computing for scientific data analyzing applications; an empirical study. In *2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 193-199). IEEE.
3. Begum, S., & Prashanth, C. S. R. (2013). Review of load balancing in cloud computing. *International Journal of Computer Science Issues (IJCSI)*, *10*(1), 343.
4. Birje, M. N., Challagidad, P. S., Goudar, R. H., & Tapale, M. T. (2017). Cloud computing review: concepts, technology, challenges and security. *International Journal of Cloud Computing*, *6*(1), 32-57.
5. Cao, Y., Wang, S., Kang, L., & Gao, Y. (2016). A TQCS-based service selection and scheduling strategy in cloud manufacturing. *The International Journal of Advanced Manufacturing Technology*, *82*(1-4), 235-251.
6. Chen, H., & Mohapatra, P. (2005). Using service brokers for accessing backend servers for web applications. *Journal of Network and Computer applications*, *28*(1), 57-74.
7. Coutinho, E. F., de Carvalho Sousa, F. R., Rego, P. A. L., Gomes, D. G., & de Souza, J. N. (2015). Elasticity in cloud computing: a survey. *annals of telecommunications-annales des télécommunications*, *70*(7-8), 289-309.
8. Danish, M., & Sharma, P. (2018). Review Study of Cloud Computing–Benefits, Risk, Challenges and Security.
9. Dastjerdi, A. V., & Buyya, R. (2014). Compatibility-aware cloud service composition under fuzzy preferences of users. *IEEE Transactions on Cloud Computing*, *2*(1), 1-13.
10. Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and engineering*, *3*(3), 227-252.
11. Hoffmann, J., Bertoli, P., & Pistore, M. (2007, July). Web service composition as planning, revisited: In between background theories and initial state uncertainty. In *AAAI* (Vol. 7, pp. 1013-1018).
12. John, H. (1992). Holland, Adaptation in natural and artificial systems, 2nd Edition ed.: The University of Michigan Press.
13. Kiruthika, J., & Khaddaj, S. (2013, September). System performance in cloud services: Stability and resource allocation. In *2013 12th International Symposium on Distributed Computing and Applications to Business, Engineering & Science* (pp. 127-131). IEEE.

14. Lan, C. W., Chen, R. C., Su, A. Y., Huang, A. F., Yang, S. J., & Chung, J. Y. (2009, October). A multiple objectives optimization approach for QoS-based web services compositions. In *2009 IEEE International Conference on e-Business Engineering* (pp. 121-128). IEEE.
15. Mukhopadhyay, D. M., Balitanas, M. O., Farkhod, A., Jeon, S. H., & Bhattacharyya, D. (2009). Genetic algorithm: A tutorial review. *International journal of grid and distributed computing*, *2*(3), 25-32.
16. Qu, L., Wang, Y., & Orgun, M. A. (2013, June). Cloud service selection based on the aggregation of user feedback and quantitative performance assessment. In *2013 IEEE International Conference on Services Computing* (pp. 152-159). IEEE.
17. Quarati, A., Clematis, A., Galizia, A., & D'Agostino, D. (2013). Hybrid clouds brokering: business opportunities, QoS and energy-saving issues. *Simulation Modelling Practice and Theory*, *39*, 121-134.
18. Sabahi, F. (2011). Cloud Computing RAS issues and challenges. *ICTer*, *4*(2).
19. Sharifi, M., Manaf, A. A., Memariani, A., Movahednejad, H., & Dastjerdi, A. V. (2014, June). Consensus-based service selection using crowdsourcing under fuzzy preferences of users. In *2014 IEEE International Conference on Services Computing* (pp. 17-26). IEEE.
20. Singh, J. (2014). Study of response time in cloud computing. *International journal of information engineering and electronic business*, *6*(5), 36.
21. Singh, S., & Jangwal, T. (2012). Price trends for cloud computing services, *International Journal of Computer Science & Information Technology (IJCSIT), 4*, 2012.
22. Sundareswaran, S., Squicciarini, A., & Lin, D. (2012, June). A brokerage-based approach for cloud service selection. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 558-565). IEEE.
23. Xiang, F., Jiang, G., Xu, L., & Wang, N. (2016). The case-library method for service composition and optimal selection of big manufacturing data in cloud manufacturing system. *The International Journal of Advanced Manufacturing Technology*, *84*(1-4), 59-70.
24. Yang, W., Zhang, C., Shao, Y., Shi, Y., Li, H., Khan, M., ... & He, H. (2014). A hybrid particle swarm optimization algorithm for service selection problem in the cloud. *Int J Grid Distrib Comput*, *7*(4), 1-10.
25. Zhang, W., Chang, C. K., Feng, T., & Jiang, H. Y. (2010, July). QoS-based dynamic web service composition with ant colony optimization. In *2010 IEEE 34th Annual Computer Software and Applications Conference* (pp. 493-502). IEEE.