Service Composition and Customization of Its Features based on Combined Classification

Mohsen Eghbali[1], Sima Emadi[2*]
1- Department of Computer Science, Yazd Branch, Islamic Azad University, Yazd, Iran.
Email: eghbali@iauyazd.ac.ir
2- Department of Computer Science, Yazd Branch, Islamic Azad University, Yazd, Iran.
Email: emadi@iauyazd.ac.ir (Corresponding author)

Received: December 2018 Revised: March 2019 Accepted: July 2019

ABSTRACT:
By promoting service-oriented architecture in e-services of organizations and inter-organizational relationships, service quality is more focused. To provide high quality combined service, it is necessary to identify quality requirements of users and offer service in line with those. Service users tend to choose a combined service among the huge collection of available services based on the quality of service. In competition among rivals, service providers must customize features of service as one of the key strategies. Customization involves the combination of service features based on user requests, these strategies raise new problems on the expression and dissemination of quality information, service identification and setting qualitative offers to service users. In the previous methods, pre-processing step was not performed in the services set, and false service suggestions to the user were possible. In this study, nearest neighbor algorithm is offered for identifying consumers and customize their quality of service. In addition, Isodata has been used to cluster and filter the services. At the end, a case study is presented to illustrate the proposed method. The results of the evaluation have shown that the proposed method has tried to solve the existing shortcomings.

KEYWORDS: Service Selection, Access Control based on The Features, Service Quality Model, Customization of Services Features.

1. INTRODUCTION

Executive web services are put on the Internet as a framework for an alliance with the distributed applications. With the increased services over the Internet, discovering the web service request is a challenging problem in service-oriented computations. In particular, the selection of appropriate high-quality web service in a large pool of functional service providers is a key factor in the selection of service. Selection decision by users is usually known by non-functional features and also as a service quality, and has been launched beyond these features. Many research activities are in the field of selection of quality-based service; in these activities, it is assumed that the service providers offer the same level of service quality for each user, and focuses on the analysis of differences or similarities between customer requirements and levels of published service quality. In the global services market with high level of competition, providing the same level of quality for all users does not express a competitive advantage to access services [1].

In the proposed method, users offer their own interests, then, the system examines users' interests with the interests of people in the past, and offers the best service to the user. Today, the combination of web services is expressed as a key challenge in web-based systems; using the appropriate method for combination leads to better and more appropriate results. With increasing research in the area of service-orientation and service combination, new and improved methods are being proposed; therefore, the necessity of the available work arises. In the following sections, selection and combination of web services based on algorithms of different categories will be discussed.

Helal et al. used the concept of replacement in service combination. In this study, after defining the service combination and its physical structure, replaceability is then defined as a standard in combination. In the methodology, at first, the qualitative model for services is selected and calculated based on the type of qualitative parameters. Using filtering technique and nearest neighbor method, a set of services are considered as candidates, then, service combination is done through genetic algorithm and a fitness function by weighted sum method. Finally, in the reprogramming phase, the failed service is replaced with regard to the replacement requirements. The advantage of this approach is application of the criteria for qualitative parameters using the weight requested by users and reducing the deviation from restrictions in the selection of service for combination. In this study, direct replacement is used, i.e. the service is replaced without any settings. So there is no compatibility between services, and it is just assumed that services are compatible with each other, which is considered as a defect in this study [2].

Another method presented in [3] is based on graph theory. In general, the main idea of this theory is that service is shown as a node. Manes are the relationships between services and their costs are in fact the quality features (here, the cost and delay). The advantages of this method include optimal runtime and memory usage. A weakness of this method is its lack of scalability.

Wu et al. used dynamic Skyline composition algorithm to combine web services based on the quality of service, where the selection of combination of web-based services is performed dynamically. In this method, with the emergence of new service, the old service is removed and the service quality is also changed. One of the advantage of this approach is the ability to identify and select the best web service from the set of services based on the quality of service, and in addition, the use of effective linear combination to reduce the number of web services selected from the set of available services. One of the disadvantages of this research is the lack of algorithm consideration in the creation of web services in the real and unreal data sets [4].

Benouaret et al. used Top-k and fuzzy logic algorithms for service preference and their combination based on user requests. In this method, work preference is modeled according to fuzzy method, and resource description framework query language is used to determine the relationship between the web services. The benefits of this study include the improvement of the diversity of web services combination along maintenance of service combinations with the highest score. The disadvantages of this research include limitation of fuzzy method service combination to match the users' requests using comparative methods [5].

In [6], using Top-K classification algorithm, combination of web services is performed to solve problems such as scalability and present the best combination to the user at the right time according to the quality parameters. The advantages of this method include scalability and using parallelism technique at the right time. The weakness of this method is the inability to respond to the user when the requested qualities are added.

By using Skyline service, Alrifai et al. reduced the number of services candidate for combination, and produced clusters of trees using K-means clustering algorithm. This algorithm receives Skyline entries, returns the binary tree structure and determines the root. In fact, it can be said that it creates a dominant relationship between web services and sorts them on the basis of the features of the service quality of service; the services belong to Skyline. The advantage of this method is the removal of invalid services, and its disadvantages include complex calculations and high processing time, which lead to reduced performance [8]. Shen et al. presented a project to optimize the combination of services aware of quality of service for multiple concurrent processes, for multiple users with different needs. They provided a protocol for reassigning service resources during concurrent execution of combined services. This method showed that its optimal features may be dynamic in an environment with limited resources. However, this project is not efficient when the number of services is high and the complexity grows exponentially [7].

Rollback is another method mentioned in the detailed methods. The main idea of this method in [8] is to solve the problem of combination of web services. This algorithm selects the desired services in a stage-wise manner; in each stage of service selection, the algorithm goes a step backwards and checks the selected services, ensuring that the best service is selected. If the selected service is approved, then it is mentioned. One of the advantages of this method is fault tolerance; when a service fails, another optimal service will be used to replace it. One of the disadvantages of this method is the increased processing time.

Klein et al. proposed a method for obtaining the optimal probability of choosing candidate services to reach the goal of long-term constraints by using a linear programming method, although QOS values were still considered to be fixed for their model. All the basic tasks aforementioned have their QOS values static and fixed for service selection. The issue of selecting component services for a combined service is under the prediction of QOS fluctuating values [9].

Schuller et al. provided an exploratory method for choosing from a set of candidate services that have randomized QOS distributions for composite services. The goal is to reduce the cost of defect in the QOS limits. They first obtained an initial selection using the ILP technique based on the fixed QOS values specified by the candidate service providers. Then, a simulation based on QOS probabilistic distributions was conducted to obtain the defect fine cost for QOS constraints, a QOS feature that is more helpful to identify flaws was identified. Finally, for each selected service w, all candidate services whose variance of their known feature is greater than w, are eliminated. The ILP once again runs on the rest of the services to select another option. The whole process is repeated until it reaches the predefined number of repetitions or the other candidate service is unavailable. Both ILP and the simulation have significant run-time [10].

As can be seen, in none of the methods, Isodata clustering is not used to filter the services and respond quickly to user requests.

The structure of the research is as follows: in the first section, the related work in the field of web services is checked; the proposed method is presented in the second section; in the third section, the resultant evaluation from the proposed method is presented; and conclusions are provided in the final section.

2. STUDY OF THE PROPOSED METHOD

Previous methods in combination and customization of services encountered problems including slow combination, high processing time, lack of scalability, etc. In [11], which was evaluated by Yan et al., in the set of services available in the data repository, the pre-processing operation was not done, and suggestion of an irrelevant and wrong service was possible; for this reason, in the proposed method in this study, the services are pre-processed, and speed and accuracy of the proposed service will be greatly improved using the clustering technique.

In the following, description of the proposed method and its solution will be discussed. Isodata Clustering method increases the speed and accuracy of service combination using division of services to different neighborhoods.

In this method, a volume that can maintain a number of $\mathbf{k_n} = \sqrt{\mathbf{n}}$ adjacent samples is created; first, the set of services is clustered into different categories according to user features, then a test data is considered with its features, and a number of K training data close to that data are analyzed in terms of the selected parameters. Following this, the user enters his favorite features, and the service is offered in accordance with the user's interests. In this study, the proposed method increases the speed and accuracy of service combination by using the division of services into different neighborhoods.

Unsupervised clustering is a fundamental tool in image processing, remote sensing techniques, and applications; this approach is useful when the available reliable data are scarce and expensive, and also when there is access to little data [12].

The approach of the nearest neighborhood using the division of services into different neighborhoods increases the speed and accuracy of the combination of services. In this method, a set is created which can hold the number $\mathbf{k_n} = \sqrt{\mathbf{n}}$ samples close to each other; at first, the set of services are clustered into different clusters according to the user's characteristics, then a test data is considered with its characteristics, and K number of training data is considered close to this data in terms of the selected parameters. After that, the user enters his or her favorite features and the service is offered in accordance with the user's interests. In this research, the proposed method increases the speed and accuracy of combining services by dividing services into different neighborhoods. It should be noted that the clustering of services is done in such a way that, according to their characteristics, the services characteristics close to each other are placed in a cluster, then, according to the request made by the user, favorite services are suggested.

Isodata clustering is an unsupervised algorithm for classification and does not need to recognize the number of clusters. Moreover, unlike K-means algorithm, this algorithm defines clusters based on user selected parameters. The algorithm divides and merges the clusters; this operation is repeated as the user reaches the desired outcome [13]. One of the widest exploratory algorithms is the Isodata clustering, in which a set of n data points are considered in the d-dimensional space together with an integer K value; also it shows the initial number of clusters and the number of additional parameters. The main purpose of this algorithm is to calculate a set of cluster centers in d-space. This algorithm is similar to K-means and aims to minimize the mean square of the difference between each point and the nearest center, which is referred to as the mean distortion. An important advantage of Isodata over K-means is that the user only needs to have an initial estimate of the number of clusters, and based on different explorations, the algorithm may propose the number of clusters by deleting smaller clusters, by integrating clusters close to each other, or by decomposing cluttered clusters [14].

The proposed method also known as combined classification and customization of services, has a suitable application when offering the best service to users. The proposed method flowchart is displayed in Figs. 4-1. First, the user inserts his request, then, all the services available in the database are called; finally, all the services called are normalized. After the implementation of the proposed algorithm, the services are customized and classified into k neighborhoods according to quality of service. Service clustering is performed in a way that according to their features, the services with close features are placed in a cluster. Then, the favorite service is offered according to the request of the user.

2.1. Call of Services Set

First, the services are called from UDDI database. Call of services is done randomly and dynamically, i.e. each time the service is called at this stage, the number of called services may be increased or decreased, but at each time of call, the qualitative features of services remain constant and unchanged. Called services have features such as availability, repetition, reliability, latency, and fee. After this stage, the services enter the filtering stage.

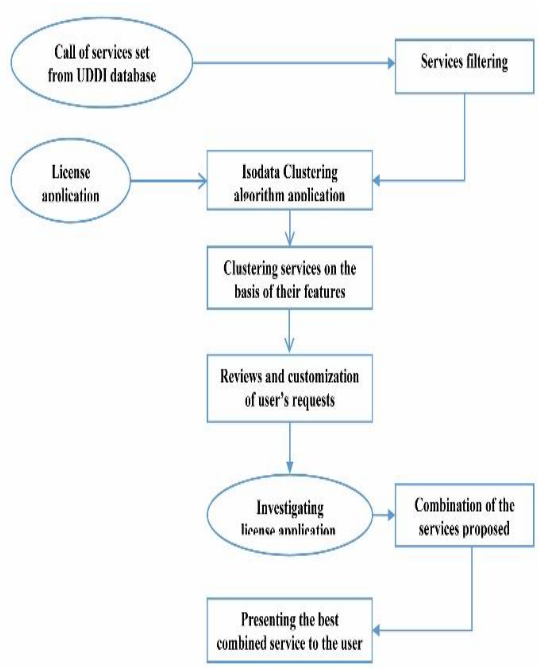

Fig. 1. Diagram of the implementation of the proposed method.

The proposed method is explained in a detailed way.

2.2. Services Filtering

After calling services from UDDI database, services are filtered. Filtering in this application means; among the services set called, incompatible services will be fully removed. Here, incompatibility means the services in which each features are not similar to the user requests in the database or the services in which features are irrelevant. For example, suppose that the data set called includes videos, and users formerly commented on each film, user comments are applied to the particular features, and the unimportant features are considered as irrelevant features. Then, in the filtering stage, only the services that meet all of the features requested are selected; services with irrelevant features are deleted. For example, assume that a set of videos is called. At this stage, any video with incomplete opinions will be identified as incompatible data and will be removed. Similarly, all services at this stage will be filtered.

In this research, the problem of clustering in multi-dimensional space occurs, and it can be identified as one of the problems of optimization of algorithms such as K-means in which the goal is to minimize the total distance to the nearest core. There are appropriate solutions for specific objectives in this regard, but there is no exact solution to determine the clustering points tangibly. There is a similar algorithm for the K-means problem that provides improvements based on the core set; but this algorithm faces difficulties for clustering problems to be implemented in real and multi-dimensional environments when K is not a small amount.

One of the most extensive heuristic algorithms is Isodata clustering where a set of n data that points in a d-dimensional space is considered with an integer K. It also shows the initial number of clusters and the number of additional parameters. The main objective of this algorithm is to calculate a set of cluster cores in d-space. The algorithm is similar to K-means and aims to minimize the mean squared difference between each point and the nearest core which is referred to as the average distortion. An important advantage of Isodata over K-means is that the user only needs to have a preliminary estimate of the number of clusters, and based on different discoveries, the algorithm may propose the number of clusters by eliminating smaller clusters or merging clusters close to each other, or by degrading the distort cluster.

Now, Isodata can run on large data sets. Due to its widespread use in remote sensing, it can have highly efficient computations. In this study, the goal is to provide a new and improved clustering algorithm to show how the proposed method can produce a more rapid implementation of the Isodata cluster. In frequent repetitions, the algorithm finds the cluster cores based on user requests, and determines merging or clustering using a number of different discoveries. At a high level, the following tasks are performed in each iteration of the algorithm: The score is assigned to the nearest cluster core, this score is based on user inputs; clusters cores are updated, and the center of gravity of related points are merged with a small number of deleted points and small clusters approved. The algorithm continues until the number of iterations is supplied more than those by the user. An example will be used to further explain this process in the following section.

As mentioned in the previous section, at first, the services are completely called and filtered, and services are selected in which all the features are applied completely. Then, the set of services are clustered according to different features. For example, consider a video set, with four features of film quality type, year of production, the production area, and the presence of specific actors, which are to be clustered. When the application is run by selecting k = 7 neighborhood, the films which have close features will be placed in a single cluster. For example, suppose that 100 films with the same year of production are placed in a cluster, 100 films with the same production quality are put in a cluster, and the same is done for other similar features in other clusters; and the films with non-close features are put in different clusters. Fig. 2 shows the services clustering based on services features.

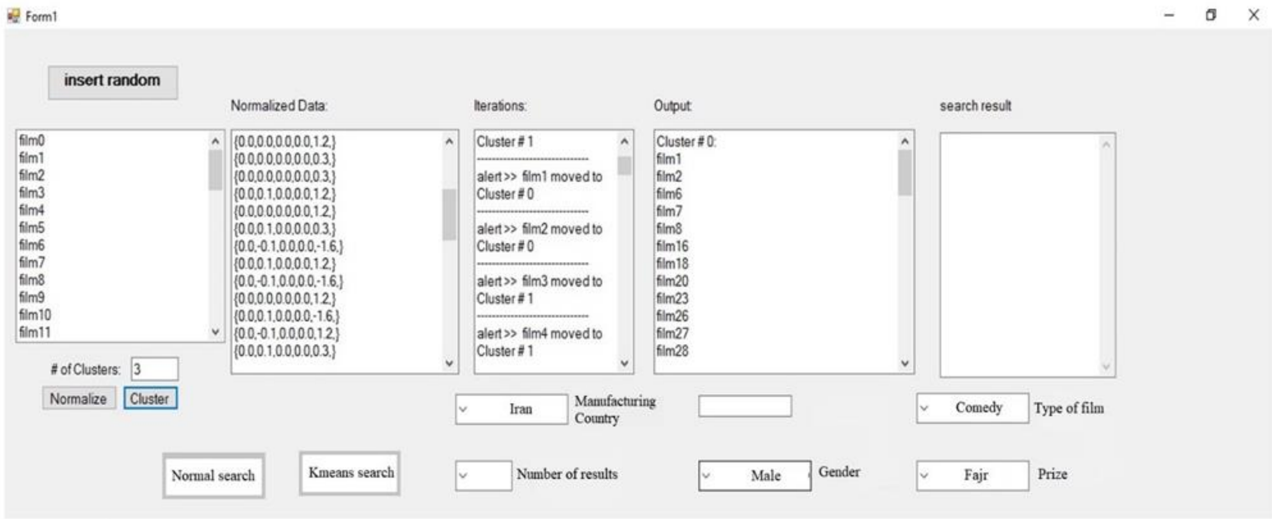

Fig. 2. Services clustering based on services features.

2.3. License Application

At this stage, some policies are applied to the services and the algorithm. The licenses will be applied at the request of the director. For example, it is assumed that the feature of gender is applied to the policy. When the user announces his/her gender as teen, only the films proper to this age are recommended and other films will not be suggested to the user. This kind of policy is only applied according to the respective director, and it will be different in different systems. An example of license application is stated in the following:

- For teens, only the films proper to this age are recommended. For example, if the teen requests for horror films that cause moral damage to him, this kind of film will not be suggested to the user.
- For an elderly person, films for children are not recommended.

Other features are expressed similarly in the following. These policies will be applied according to the director.

2.4. User's Request and Customization

At this stage, the user's request is announced according to his favorite features. Then the request is examined in various clusters, and cases in accordance with the interests are suggested to the user. For example, the user's interests include videos for the age group of: teenager, quality: Full HD, presence of a certain actor, year of production: 2005, and area of film production: Hollywood. These features are considered in various clusters and some films will be offered to the user. These films are selected and recommended to the user according to the permits previously applied.

2.5. Investigation of the License Application

At this stage, applied permissions and policies are investigated; after examination of the licenses, user's request will be customized, and the user receives some services that are tailored to his interests. For example, the user inserts his favorite video features and all clusters are investigated for each feature, and at the end, determining the number of requested videos, the user asks for the best suggestions. For example, the user imports the features of film model, year of production, gender and other cases. The user requests some suggested films, and the best videos will be offered to the user.

2.6. Combination of the Proposed Services

After proposing the user's requested services, the user announces a request based on the best combination of the proposed services. Then, the final proposal will be checked according to the requested features. Services are selected and combined according to the user's interests, and at the end, the best combination service will be offered to the user. Suppose that a user imports his favorite video features as age groups of: young, year of production: 2010, country: Iran, type of award: Award of Fajr, genre: comedy and other features. Then, from the recommended videos, features of the films will be, respectively examined in accordance with the interests requested by the user. The methodology is such that the film features are compared using logic of and and or. The video whose features are the same as the user's requested interests is 1, and if the features are not the same, it is 0. Accordingly, the films are considered along with the user's feature and request. Finally, some films are suggested to the user, the films which perfectly match the user's interests and other films that are not in accordance with the user's interests, will not be recommended.

Web service combination pseudo-code for selection and customization of the service is as follows:

```
for (int k = 0; k < _numberOfClusters; ++k)
          {
              distances[k] =ElucidanDistance(_normalizedDataToCluster[i], _clusters[k]);

              if (newClusterId != _normalizedDataToCluster[i].Cluster)
             {
                   changed = true;
                   normalizedDataToCluster[i].Cluster=
                  _rawDataToCluster[i].Cluster=
                   newClusterId sb.AppendLine("alert >>  " + _rawDataToCluster[i].Name +" moved to Cluster # " +
                       newClusterId);
             }
             else
           {
                   sb.AppendLine("No change.");
            }

          }
```

3. SIMULATION RESULTS OF THE PROPOSED MODEL

To analyze and evaluate the proposed method, the approach offered by the conventional proposed method in [6] was evaluated first, and then, the results obtained with the same data on the proposed algorithm were evaluated. In this implementation of data set, the desired services are created by the program. The system used is Sony Corporation motherboard, Intel (R) Core (TM) i5 CPU, and Microsoft Windows 10 operating system. The services are created randomly by the program in the range of 50-items. After each call, the qualitative features of services are static. The static quality means that the qualitative features of services are fixed per call, and they are not changed at per call. However, in the real environment, these features may be dynamic or flexible in the database.

Results of the nearest neighborhood algorithm in the proposed method and customization of services set will be explained in the following.

3.1. Call of Services Set

In the analysis of the results obtained, it is observed that with increased number of web services in the proposed method, runtime is increased. Then, program runtime is measured with the number of web services upper and lower scale. The beginning time of the program is investigated with k = 7 clusters, then k = 11 and k = 15 clusters.

Below, in Figs. 3 and 4, an example of the diagrams obtained for the neighborhood of k=7 at the scale of lower and upper number of services is shown.

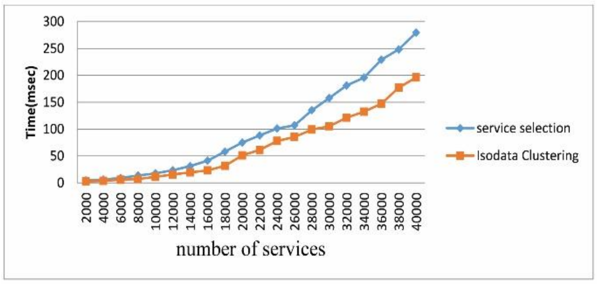

Fig. 3. Comparison of the services offer time at lower scale with 7 clusters.

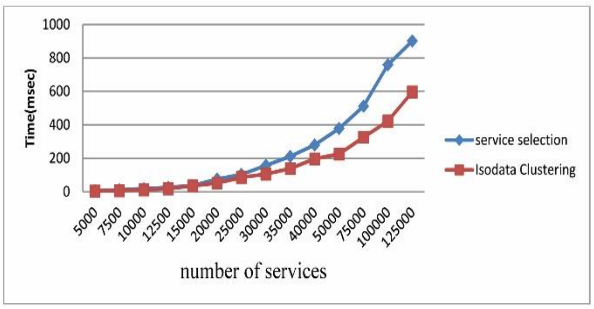

Fig. 4. Comparison of the services offer time at upper scale with 7 clusters.

Then, comparison of the services with neighborhood of k=11 is also shown in Figs. 5 and 6.

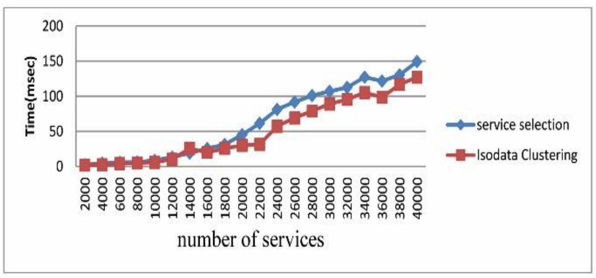

Fig. 5. Comparison of the service offer time at lower scale.

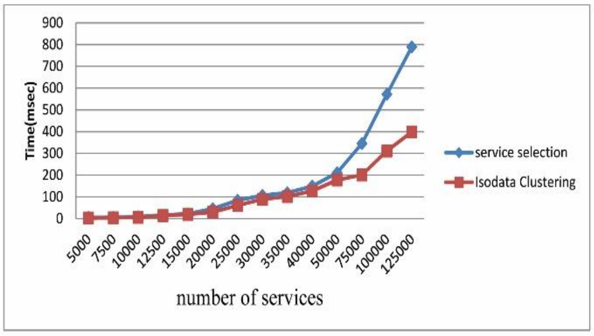

Fig. 6. Comparison of the service offer time at upper scale.

As shown in Figs. 3-6, with a significant increase in the number of services, the response time of the proposed method is significantly different from the one presented in [6].

In addition, the comparison for k=15 is shown in Figs. 7 and 8. As shown in Figs. 7, 8, with a significant increase in the number of services, and with 15 clusters, the response time of the proposed method is much better than the proposed method in [6].

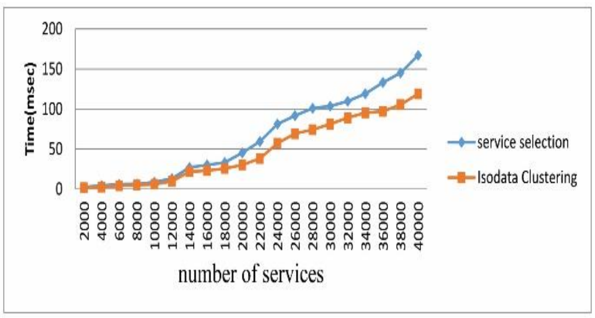

Fig.7. Comparison of the offer time for 15 clusters at lower scale.

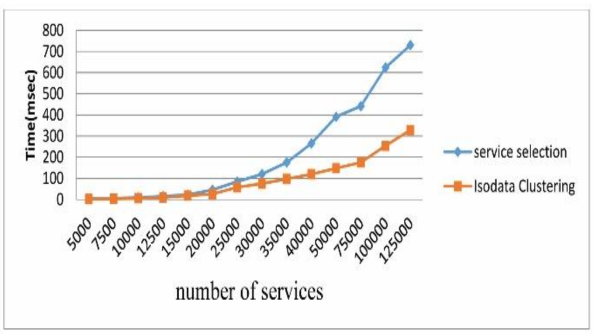

Fig. 8. Comparison of the offer time for 15 clusters at upper scale.

In the following, the times suggested in each cluster are compared. As shown in Fig. 9, more clustering reduces the proposed time and increases the speed of action.

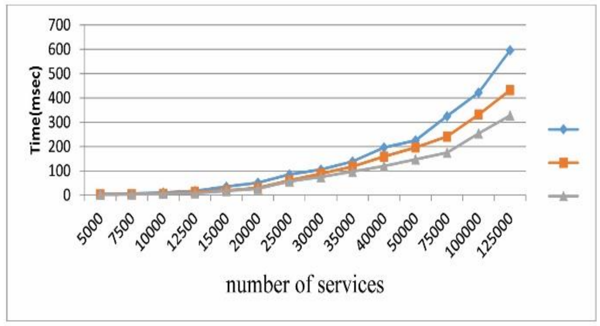

Fig. 9. Comparison of the recommended times for different clusters.

The graphs shown in Fig. 9 conclude that as the number of clusters increases, the time of service composition decreases. This reduction in time increases the speed of the combination.

3.2. Memory Usage

The other considerable case that can be analyzed in the results obtained is the memory usage by the proposed algorithm. After calling the services set, they enter the phase of normalization. After normalization process, consumption volume of the services is reduced which will lead to lower system cache and faster program processing. In Fig. 10, memory usage is investigated in three different clustering.

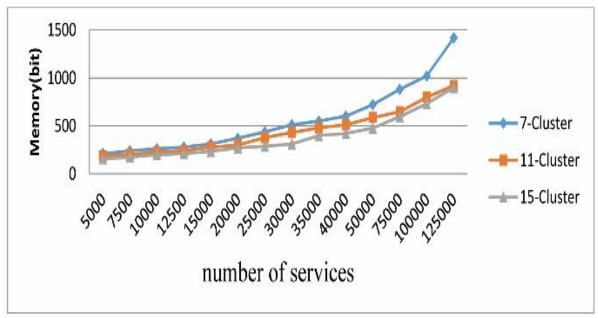

Fig. 10. Comparison of the memory usage in different clustering.

4. CONCLUSION AND FUTURE WORKS

Increasing web services result in services with the same capabilities and competition between them. The choice of appropriate service with regard to user requirements is very important; customization of services is possible through Isodata Clustering algorithm. With this algorithm, which is an improved K-means algorithm, customization of web services is possible. The proposed algorithm reduces the computation for the combination of Web services. This method, by using filtering and clustering method, reduces the time of selecting services, and this reduction in time increases the speed of the combination of services. In addition, for better a performance of the algorithm to the client, several optimal responses are suggested to the customer according to previous feedbacks. The function of the proposed algorithm is designed in such a way that it can receive other offers if the user does not have the full satisfaction of the combined service provided.

REFERENCES

[1] Soltan aghaie, M., & Khatibi, N., "**Study of Different Methods and Compare the Different Approaches that Combine Services,"** Conference on Science and Computer Engineering Islamic Azad University of Najafabad., 1391.

[2] Al-Helal, H., & Gamble, R. **"Introducing Replaceability into Web Service Composition".** IEEE Transactions on Services Computing,**"** Vol. 7(2), pp. 198-209, 2014.

[3] Pejman, E., Rastegari, Y., Esfahani, P. M., & Salajegheh, A. **"Web Service Composition Methods: A survey."** In Proceedings of the International Multi Conference of Engineers and Computer Scientists Vol. 1, 2012.

[4] Wu, J., Chen, L., & Liang, T. **"Selecting Dynamic Skyline Services for QoS-based Service Composition**," Applied Mathematics & Information Sciences**,** Vol. 8(5), 2579, 2014.

[5] Bock, D. E., Mangus, S. M., & Folse, J. A. G. **"The Road to Customer Loyalty Paved with Service Customization."** Journal of Business Research, Vol. 69(10), pp. 3923-3932, 2016.

[6] Yan, J., Gao, H., & Mu, Y. **"Attribute Based Service Customization and Selection."** In 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, pp. 57-64, 2014.

[7] Memarsadeghi, N., Mount, D. M., Netanyahu, N. S., & Le Moigne, J. "**A Fast Implementation of the ISODATA Clustering Algorithm."** International Journal of Computational Geometry & Applications, Vol. 17(01), pp. 71-103, 2007.

[8] Zhang, W., Chang, C. K., Feng, T., & Jiang, H. Y. "**QoS-based Dynamic Web Service Composition with Ant Colony Optimization."** In 2010 IEEE 34th Annual Computer Software and Applications Conference, pp. 493-502, 2010.

[9] Shen, Y., Yang, X., Wang, Y., & Ye, Z. "**Optimizing QoS-aware Services Composition for Concurrent Processes in Dynamic Resource-Constrained Environments,"** 19th International Conference on Web Services (ICWS), pp. 250-258, 2012.

[10] A. Klein, F. Ishikawa, and S. Honiden, **"Efficient QoS-Aware Service Composition with a Probabilistic Service Selection Policy,"** in Service-Oriented Computing**.** Vol. 6470, P. Maglio, M. Weske, J. Yang, and M. Fantinato, Eds., ed: Springer Berlin Heidelberg, pp. 182-196, 2010.

[11] D. Schuller, U. Lampe, J. Eckert, R. Steinmetz, and S. Schulte, **"Cost-Driven Optimization of Complex Service-Based Workflows for Stochastic QoS Parameters,"** in 2012 IEEE 19th International Conference on Web Services (ICWS'12), pp. 66-73, 2012.

[12] Benouaret, K., Benslimane, D., Hadjali, A., & Barhamgi, M. "**Top-k Web Service Compositions using Fuzzy Dominance Relationship,"** IEEE International Conference on Services Computing (SCC), pp. 144-151, 2011.

[13] Deng, S., Huang, L., Tan, W., & Wu, Z. "**Top-automatic Service Composition: A Parallel Method for Large-Scale Service Sets."** IEEE Transactions on Automation Science and Engineering, Vol. 11(3), pp. 891-905, 2014.

[14] Memarsadeghi, N., Mount, D. M., Netanyahu, N. S., & Le Moigne, J. "**A Fast Implementation of the ISODATA Clustering Algorithm."** International Journal of Computational Geometry & Applications, Vol. 17(01), pp. 71-103, 2007.