http://www.jscdss.com
Vol.4 No.3 June 2017: 21-26
Article history:
Accepted 29 June 2017
Published online 29 June 2017

Journal of Soft Computing and Decision Support Systems

E-ISSN: 2289-8603

Web Service Composition Based on Qualitative Parameters Using Top-K Color Classification

Elahe Hajaliasghar Ajil Frosh [a], Sima Emadi [b]
[a] Department of Computer Engineering, Yazd Branch, Islamic Azad University, Yazd, Iran
[b] Department of Computer Engineering, Yazd Branch, Islamic Azad University, Yazd, Iran

*** Corresponding author email address**: emadi@iauyazd.ac.ir

Abstract

Service-oriented architecture is the key to run web services. It has been a promising solution for composing services. Given the increase and complexity of the users' needs, atomic services will not be adequate in service-oriented systems. Thus, drawing on the ability to compose services, scholars are searching for an optimal composition with identical efficiency among web services. Of course, service composition involves attention to the users' needs and adapting them to qualitative parameters as the non-performance requirements of the system during the composition. The composition of automatic services will be a useful technology in composing multiple atomic services; however, the present attempts do not bring about the favourable efficacy. Many methods have been proposed by the researchers for composition of automatic services. However, in this paper an effort has been made to ameliorate the defects by introducing changes in Top-K algorithm. In the proposed method, the concept of 'color' has been employed in the section dealing with service composition in order to discard the backtracking technique in Top-K algorithm and the objective is to increase efficacy in composing web services and improve the time in the execution of the composition process. The proposed method improves the rate and accuracy of algorithm execution by displaying the services using graphs and the implemented color. Compared to the existing methods, the evaluation of the recommended method indicates a drop in memory usage, an increase in the speed of algorithm execution and the achievement of optimal results in less time.

Keywords: Web service composition, Classification algorithm, Parallelization, Top-K algorithm

1. Introduction

A great deal of attention has been focused on service-orientation with regard to the capacity of being used in various places independently of any particular technology on a large scale by viewing the user's varying needs (Liangzhao et al., 2006).

In some cases, as the atomic service is not efficient in responding to the user and the user's needs are not met, composing atomic services and arriving at an optimal service composition that has the quality that the user requires are prominent issues in service-orientation (Liangzhao et al., 2006). Given the cases that were mentioned, the objective in automatic service composition is to provide new value-added services from the ones that exist, thus leading to more efficient user services (Liangzhao, et al., 2006). To select from a large number of services, there are various methods at the level of web compositions and an optimal composition can be provided considering qualitative parameters which indicate the survey of services (Liangzhao, et al., 2006). If several different services are available to achieve a certain goal, the user can make a choice depending on his or her demand to select the best composition which is possible.

In sum, an increase in the number of services, the existence of a bottleneck and scalability are among the serious challenges that should be regarded in service compositions. Despite other methods, the recommended method allows the users to access diverse compositions to meet their preferences. Therefore, if an optimal composition is invalid, another composition can be chosen as an alternative. Having access to additional compositions is very effective at the time the algorithm is being executed in that the user will not need to apply the algorithm once again to find a new composition. Top-K algorithm increases the consumable memory due to the application of the backtracking technique in finding the courses.

Wei et al. (2014), proposed a method by applying the genetic algorithm to select the best service while composing the service from similar ones with regard to the applicable programming and all accessible qualitative parameters. The best service composition based on applicable programming and high scalability can be mentioned as the advantages of this method. One of the weaknesses of this method is the absence of parallelization technique similar to Top-K in the process of composition which leads to a faster composition.

The other proposed method (Pejman, et al., 2012) is based on the graph theory. In general, the central notion in this theory is that the service has been shown as a knot. The edges express the connection between services. Actually, the expenses of the edges are quality features (i.e. cost and delay). The optimal execution time and the level of consumable memory are the advantages of this method. One of the weaknesses of this method is the lack of scalability.

Safi et al. (2013) performed a composition process of all existing services. This research was performed by focusing on web service composition in order to reduce the time of execution, the general cost of processing, and achieve better efficiency compared to similar methods. Consequently, based on the comparison between this project and previous research, the conclusion was drawn that the project efficiency is considerably high and useless web services are non-existent in the results. The defects of this method are the lack of parallelization in the service composition process and space shortage for lists and tables.

In comparison with previous methods, Navarro et al. (2012) proposed an optimal design in an article at the time of *O* (*K*) and in *O* (*N*) linear space. This method places the values in a descending order with regard to the qualitative parameters of web services. A tree, whose nodes each have a color, specifies Top-K values in a descending order and in response to the search on the course between two knots. The lack of parallelization for the faster presentation of a suitable composition and the problem with the consumable memory are among the disadvantages of this method (Navarro et al., 2012).

Top-K query algorithm in (Deng et al., 2013; Zhang et al., 2010) has been provided on the basis of KPL algorithm. In this algorithm, a DAG graph is utilized; as for the results of service composition, an independent graph has to be created first, and the user is then asked to provide a service according to Top-K algorithm. This design enhances the scalability and accuracy.

Wagner et al. (2011) & Shen et al. (2012) was employed a data structure for services with a similar performance so that they could be set in one cluster. In this research, service quality is computed for each cluster and the values of each cluster are viewed in a composition process to collect them for general computations in a separate section. Preliminary research has separately neglected the non-performance features of each service in automatic service composition. In addition, the quality of each cluster has been taken into account in general computations. The results of service composition are normally done using few services.

The article structure is in the following manner. In the Section 2, relevant works on service composition will be presented and the recommended method will be discussed in the Section 3. In Section 4, as regards the results derived from the recommended method, the analysis and assessment are presented and the conclusions and future recommendations are provided in Section 5.

2. The recommended method

In the proposed design, a novel method has been examined to improve the efficiency of Top-K algorithm. This algorithm has been implemented based on the backtracking technique in the acyclic graph design and the depth-first search. With this technique, all the solutions are established given the concepts derived from web services. Another problem with Top-K algorithm is the use of stack in composition. In the proposed method, colors have been assigned based on the qualitative parameters of services at each stage of web service composition to deal with the problem of overuse in memory in Top-K algorithm. The changes that will be enumerated in the following pages have been attempts made to optimize the ways of arriving at the composition results.

In this article, an innovative method called Top-CWSC has been provided: The "Top-CWSC method" has filtered the services on the outputs of Top-K algorithm.

2.1 The pre-process of raw data

In the pre-process phase, a raw dataset with a large scale is processed to be transformed into a rule reservoir. In other words, it can be asserted that the services of the combined data are transformed into rules to create a rule reservoir. The rule reservoir is a kind of data structure in memory that can be quickly accessed in response to the users (Hennig et al., 2010).

A rule reservoir is produced based on Tables 1-3. A rule has been defined as 4-tuple ⟦r=<S,C ⟧_I,⟦C ⟧_O,QoS> whose parameters are as below (Hennig et al., 2010):

i. S denotes the service producing the rule.

ii. CI denotes a set of concepts that are required by the rule as input which includes the web service inputs.

iii. CO denotes a set of concepts that are concluded by the rule as output including the web service outputs.

iv. QoS indicates the characteristics of the rule quality utilized in the implementation and the response time has been regarded as the service quality parameter in this research. Each web service can be transformed into an r rule in which ⟦r.C ⟧_I is an equivalent of ⟦S.C ⟧_I and ⟦r.C ⟧_o, ⟦S.C⟧_o is the conjunction and ⟦S.C ⟧_o denotes previous concepts. This means that a rule can create the concepts known in a service as well as their support. Once the services all transformed into rules, a rule reservoir was made. The reservoir saves the data in the memory of more than one database, thus enabling a fast and efficient access while the composition is being established.

JSCDS An International Journal
E-ISSN: 2289-8603

2.2 The implementation of Top-K algorithm

Filtering the services. When a request is received from the user, the services are formed in the rule reservoir and those services that are not useful in the final results are filtered owing to the user's request and the existing rules in the reservoir. On a large scale of web services, in reducing the number of services chosen for the composition, the services that do not have the same input and output are omitted. In the proposed method, this technique has been implemented by pruning similar services. From these similar services, only the service that has a better priority will be selected. After the pruning, the filtering of useless services in Top-K algorithm is as below (Hennig et al., 2010):

i. The input set is created using primary elements including the inputs in the user's request.
ii. All the services whose inputs contain the input elements are found and their output is then added to the input set and the services are preserved.
iii. Phase 2 is repeated to such a degree that a service whose inputs constitute the existing elements in the input set is no longer found.

Table 1
A sample of rule reservoir

Number of services	C_I	C_O	Cost
Service 1	A	D	125
Service 2	E	D	183
Service 3	D	B	55
Service 4	D	C	86
Service 5	F	C	110

Layout. The data size grows smaller due to the pruning of a number of web services that exist in the rule reservoir in the beginning of the process. Afterwards, the remaining services that are not commensurate with the user's request will be discarded though filtering. Consequently, web services are taken from the datasets and the layout of services is achieved according to (Fig. 1). As mentioned earlier, in Top-K method, the useless services that do not provide the user with the required concepts are discarded in the filtering phase.
Graph design. First, contrary to the Top-K method, the forward search is done from the vertex with respect to the user's request.
Graph ranking. At each level, regarding the values of qualitative parameters as determined by the user, the knots are included in the graph. In the proposed method, three colors have been considered as the basis of user's parameters.

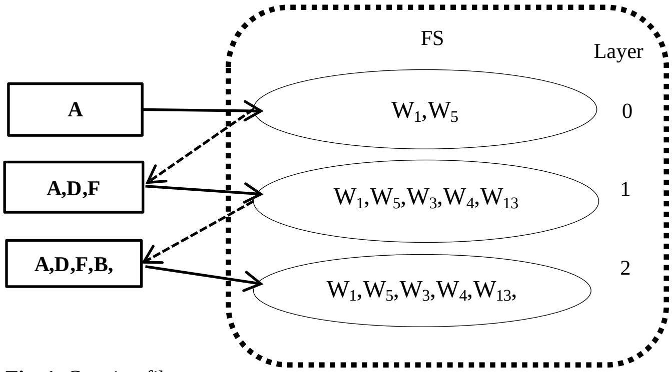

Fig. 1. Creating filters.

Table 2

The general service quality values

Service	Input1	Input2	End1	End2	All QoS	Cost
Service 1	A		D		43	125
Service 5	F	A	C		54	110
Service 3	D		B		34	55
Service 4	D		C	F	45	86
Service 11	B		F		32	88
Service 13	E	F	A		65	110

Color assignment to web services. A table is used to save the quality of optimal service value and the service parents. There might be multiple services having the same general service quality value for the same inputs. The existing services are prioritized based on the value of general service quality. This general amount of qualitative parameters is used to assign colors to services in the graph. Each level of the graph is classified into three groups to be used as a basis for color assignment. The significance of the table lies in the computational mode of general qualitative parameters for each service.
The computation of general service quality. The cost of each node must be calculated at the level of each graph. The calculation of the cost at this stage is based on the implemented qualitative parameters in the proposed method. These parameters are: cost, response time, reliability, and attraction. The selection of one or more qualitative parameters depends on the user's needs and is

E-ISSN: 2289-8603

implemented in accordance with his or her view. Therefore, the nodes at each level should be classified regarding the specified number of colors, while colors would be assigned to services afterwards. The service that has an optimal qualitative value beside other services will be assigned the color with a better priority. The color assignment will continue in the same order for the remaining services. These phases will be repeated until the services are finished at the final vertex. Having examined the whole course, the composition results are derived. The section dealing with composition which comprises the total cost calculation and the method for doing these calculations will not differ in any of the two proposed methods.

Table 3
The computation of total service quality

Services	CO	All QoS
Service 1	**D**	**43**
Service 5	**C**	**54**
Service 3	**B**	**34**
Service 4	**C**	**45**
Service 11	**F**	**32**
Service 13	**A**	**65**

As an example, from the steps in executing the algorithm, the final graph used to find the best composition within the shortest space of time has been illustrated in (Fig. 2).

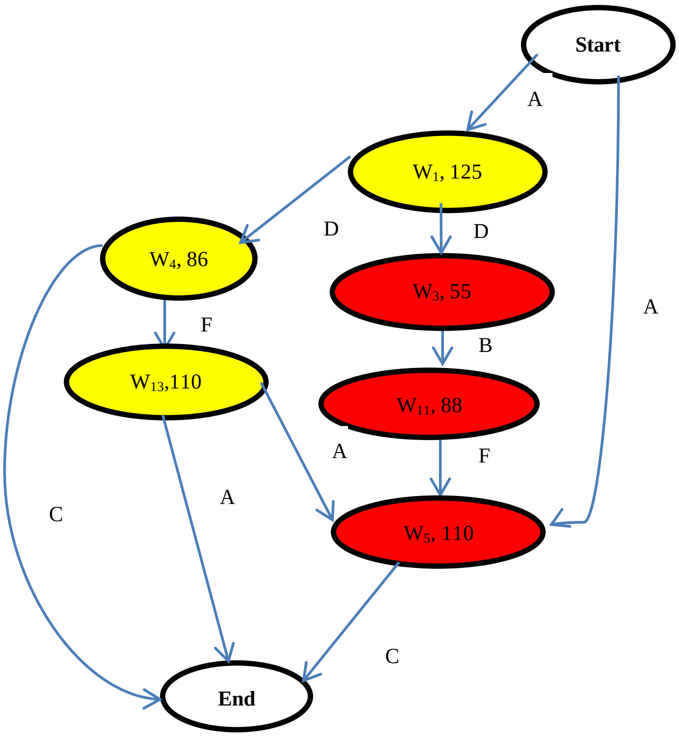

Fig. 2. Creating filters.

3. Investigating the results of the proposed model simulation

To analyse and evaluate the Top-CWSC method, the implementation of the presented approach was initially examined by Deng and the results were then analysed by identical data in the provided algorithm and the recommended methods. In this implementation, the program establishes the dataset of the services. The system that is used consists of Motherboard Asus Tek N82JQ, Intel(R) Core (TM) i7 processor, and Microsoft Windows 7 Ultimate 32 Bit system.

The application results of Top-CWSC proposed methods in eliciting multiple optimal compositions from the web services are as follows: with regard to the fact that web service compositions have been performed using the filtering phase in Top-K algorithm and Top-K Color in the proposed methods, the time consumed in filtering identical web services has been remarkably diminished by using Top-CWSC compared to method (Deng et al., 2014).

Memory usage. Another noteworthy fact which can be analyzed in the results is the level of memory usage in the proposed method. Bearing in mind that in the recommended filtering method, the number of selected services for composition has decreased (Fig. 3, Fig. 4, Fig. 5, and Fig. 6), the memory level which is consumed is therefore less compared to the Top-K method and a desirable improvement appears in memory usage.

JSCDS E-ISSN: 2289-8603

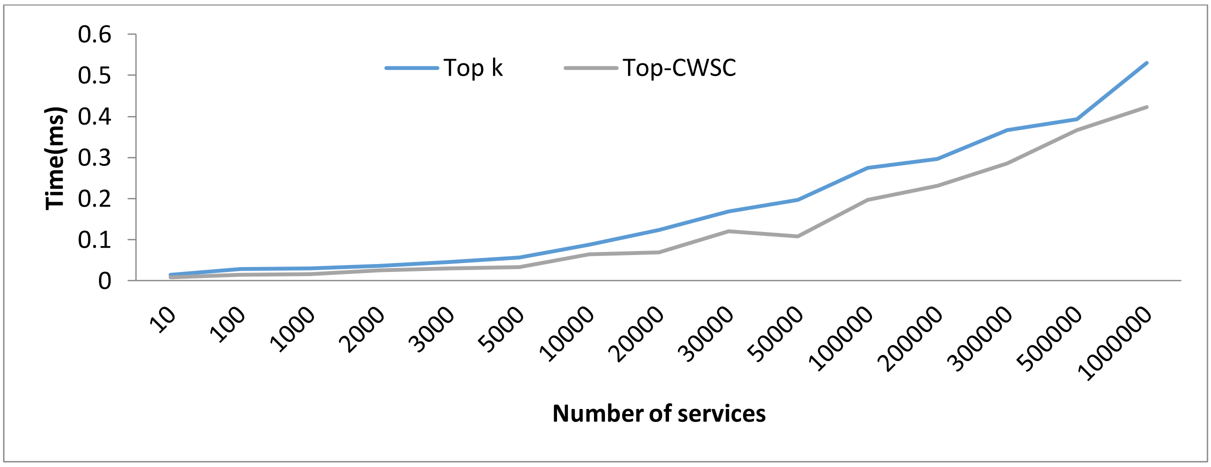

Fig. 3. The comparative diagram of execution time based on cost.

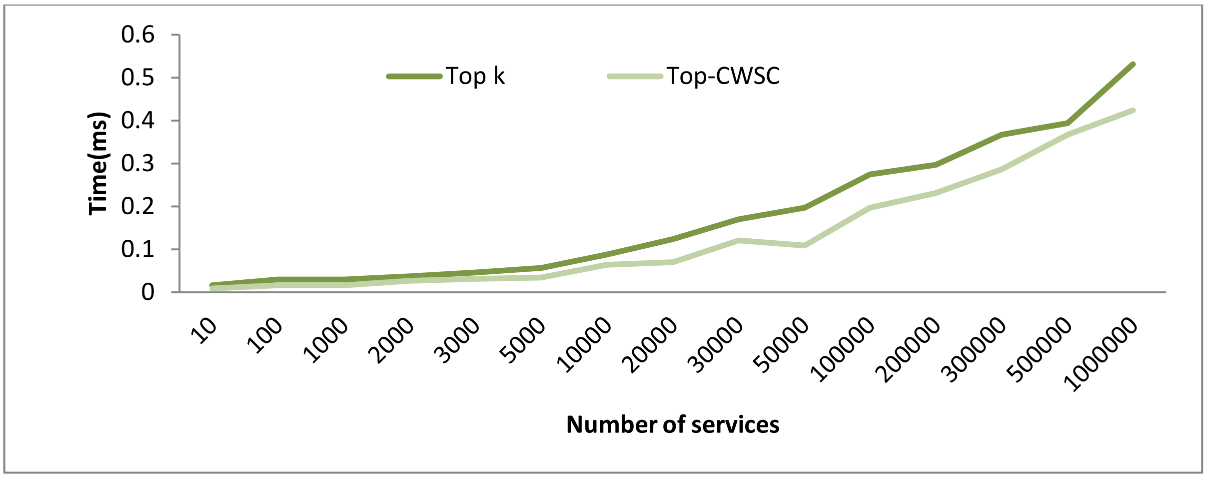

Fig. 4. The comparative diagram of execution time based on response time.

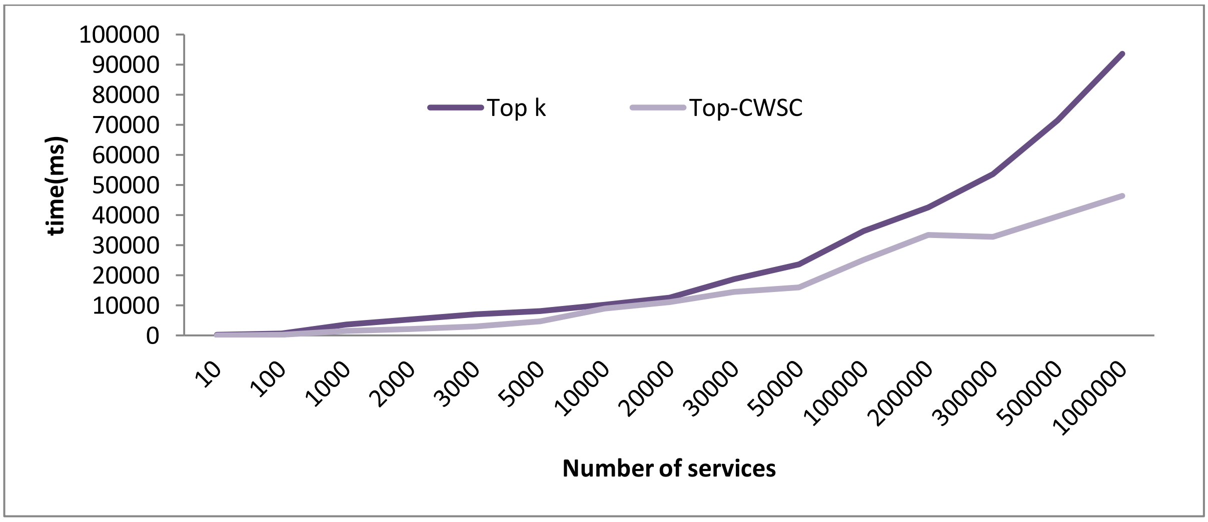

Fig. 5. The comparative diagram of memory usage based on costs.

E-ISSN: 2289-8603

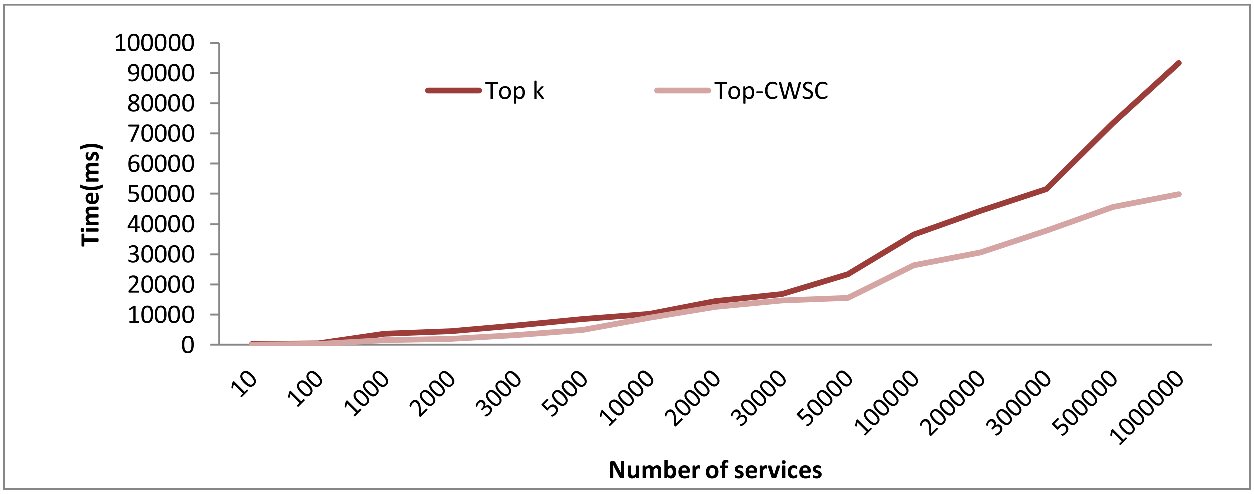

Fig. 6. The comparative diagram of memory usage based on response time.

4. Conclusion and future works

In recent years, the number of web services with similar performance and different quality has been on the rise. With the expansion and improvement of web services, the scholars presented algorithms for automatic service compositions that not only provide the correct composition operation for a huge number of services, but also meet the quality needs of the users. In the extant methods, presenting only one solution in certain conditions is not optimal in terms of flexibility and diversity. In this research, a method was proposed that assigned colors based on the qualitative parameters of the services to avoid the problem of 'memory overuse' in Top-K algorithm at the composition stage. As for the recommended future works, it is possible to point to the replacement of services when the existing services are not used and also the employment of real data in service composition which might be very effective in providing the ultimate response for the user.

References

Deng, D., Huang, S., Tan, L., & Wu, Z. (2014). Top-Automatic Service Composition: A Parallel Method for Large-Scale Service Sets. IEEE Transactions on Automation Science and Engineering, 11(3), pp. 891-905.

Deng, S., Wu, B., Yin, J., & Wu, Z. (2013). Efficient planning for top-K Web service composition. Knowledge and information systems, 36(3), 579-605.

Hennig, P., & Balke, W. T. (2010, July). Highly scalable web service composition using binary tree-based parallelization. IEEE International Conference on In Web Services (ICWS), (pp. 123-130).

Jiang, W., Hu, S., & Liu, Z. (2014). Top k query for QoS-aware automatic service composition. IEEE Transactions on Services Computing, 7(4), 681-695.

Pejman, E., Rastegari, Y., Esfahani, P. M., & Salajegheh, A. (2012, March). Web service composition methods: A survey. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1).

Navarro, G., & Nekrich, Y. (2012, January). Top-k document retrieval in optimal time and linear space. Society for Industrial and Applied Mathematics. In Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms (pp. 1066-1077).

Safi, A., Jawawi, D. N., & Wakil, K. (2013, December). Web services composition with redundancy consideration. IEEE Conference on Open Systems (ICOS), (pp. 112-117).

Shen, Y., Yang, X., Wang, Y., & Ye, Z. (2012, June). Optimizing QoS-aware services composition for concurrent processes in dynamic resource-constrained environments. IEEE 19th International Conference on In Web Services (ICWS), (pp. 250-258).

Wagner, F., Ishikawa, F., & Honiden, S. (2011, July). QoS-aware automatic service composition by applying functional clustering. IEEE International Conference on In Web Services (ICWS), (pp. 89-96).

Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J., & Chang, H. (2004). Qos-aware middleware for web services composition. IEEE Transactions on software engineering, 30(5), 311-327.

Zhang, W., Chang, C. K., Feng, T., & Jiang, H. Y. (2010, July). QoS-based dynamic web service composition with ant colony optimization. IEEE 34th Annual. In Computer Software and Applications Conference (COMPSAC), (pp. 493-502).

.

JSCDS
E-ISSN: 2289-8603