

QOS CORRELATION WITH CONSTRAINT BASED OPTIMAL SERVICE SELECTION IN COMPOSITION

Yashwant Dongre^{1*} and Rajesh Ingle²

¹Pune Institute of Computer Technology, Pune. Savitribai Phule Pune University, Pune, India.

² Pune Institute of Computer Technology, Pune. Savitribai Phule Pune University, Pune, India.
¹yashwant.dongre@gmail.com, ²ingle@ieee.org

Abstract: *In service-oriented architecture, service composition plays important role as key technology for integrating complex web applications. The service selection as part of service composition process attracted towards Quality of Service as mandatory criteria for optimal selection of services for composition process. While fulfilling the non-functional requirements from the user among functionally similar list of services is very challenging task. Web services are treating Software as a Service for building web applications. The Quality of Service correlation of services with constraint for selection of service is proposed in this paper. Experimental results of correlation without constraint method and proposed method shows that our method outperforms over existing approach.*

Keywords: Service selection, Quality of service, Optimization of problem, QoS correlation.

1. INTRODUCTION

Over the current two decades, publication and deployment of web service is increasing over distributed networks. A web service is functional module realized by web accessible programs, database, sensors and variety of other physical devices on the web and is identified by Uniform Resource Identifier (URI) [1].

In the service computing environment selecting most relevant web service to compose complex application is challenging activity. At the same time fulfilling requirement of end user is also mandatory task. Now days Internet of Things (IoT) technology is demanding in the form of web service, hence so many providers are providing IoT functionalities as Web services. Using standard network technology web service exports description of applications' functionality which makes it distributed over network. For the execution of business process web service provides distributed environment [2].

For every similar service functionality there is large number of services available differing in quality of service. The number of candidate service belongs to abstract task which handles functionally similar services respective to each task. Abstract task is nothing but service class which is having same functionality of service in group [3][4].

In the process of service composition, services selection is one of the tasks among many. The selection task needs selecting one of the services from list of functionally similar services with varying QoS, due to this is very difficult. QoS parameters [5] are like response time, reliability, availability, reputation, execution time, cost, and price etc. QoS of service are interrelated, interdependent and conflicting with each other. More alternate choices of offer from service providers due to the increased number in services to the user [6]. But due to this increased number in services available, selecting suitable best web service to address issue in task become difficult activity. In this case users force the QoS to select the best service described by providers which meet non-functional requirement for composing application.

The rest of the paper is organized as: Section 2 briefs the related works. The problem definition is explained in Section 3. The description of proposed approach is presented in Section 4. Experimentation is presented in Section 5. Results and discussion are briefed in Section 6. And at last Section 7 is about conclusions.

2. RELATED WORK

This section is divided into Correlation aware works and non-Correlation works

2.1. Correlation aware selection

F. Wang et. al. in [7] considered two typical correlations, compensability-oriented correlation and quality-oriented correlation. The feasibility of solution depends on compensability-oriented correlation. The influence of correlation between two services on the overall QoS of a composition is quantified by a discount percentage. Authors in this works presented mathematical model for correlation-aware services. This work designed mimetic meta-heuristic based algorithm called Hyper volume Estimation Algorithm. To achieve better trade-off among multiple conflicting QoS criteria, correlation-based local search strategies are also presented.

D. Li et. al. in [8] proposed service selection approach, which considers QoS correlations of services and user requirements. To avoid the single point of failure, approach used decentralized topology.

In [9] S. Deng et. al. presented correlation aware service pruning for service selection. This method prunes all services that are not optimal candidate services. In this work authors claim they can manage complicated correlation of services and shows improvement in QoS values of composite services.

In [10] Y. Du presented approach to perform QoS based correlations using service skyline with single dimension. This work author assumed candidate services for selection per task have no dependence with services in other tasks. To reduce search space some advanced pruning algorithm with service lattice and a min-heap is proposed based on service skyline.

2.2. Non-correlation aware selection

L. Purohit et. al. in [11] proposed an approach for QoS service selection. This method used classification technique for pre-filtering of candidate web services. Also, PROMETHEE Plus, is applied for eligible services. Hybrid weight evaluation mechanism is used for Maximizing Deviation. The services matching with the QoS requirements of the end user are selected.

W. Wang et. al. in [12] proposed service selection approach to avoid the unreliable real-world process of obtaining services. To enhancing the reliability two-phase decision is used. For solving the optimization problem, a convex hull-based approach is adapted. Large-scale and real-world experiments are conducted to show the advantages of the proposed approach.

R. Z. Yasmina et. al. [13] proposed a framework that allows selecting the Top K compositions of services that best meet the user's requirements. This framework handles the user's global constraints and fluctuating nature of the QoS information. Authors present rank the services of each abstract class according to the probabilistic dominance heuristic. Authors also present explore the compositions search space by leveraging the backtracking search.

A. S. Kurdija et. al. [14] proposed service selection algorithm using heuristic method for multiple criteria., The proposed method reduces the problem to transportation problems. The cost based utility function is is used to improve solution.

M. Moghaddam et. al. in [15] extended service selection to address composite service requests. Authors propose selection mechanism based on combinatorial auctions to solve this problem. The results in work show that the proposed mechanism attains a good success rate in finding solution.

M. E. Khanouche [16] proposed teaching learning-based QoS-aware services composition algorithm is proposed to find a very close to the optimal composition in a reasonable amount of time. Unlike the evolutionary computation-based and swarm intelligence-based approaches, the proposed services composition algorithm is characterized by a few tuning parameters and a high exploration capability of the search space of compositions. This allows obtaining compositions with a high optimality in terms of QoS and without requiring some hard-tuning parameters. The simulation results show clearly that algorithm performs better in terms of the composition optimality and has a reasonable execution time in a large-scale environment.

C. Hang et. al. [17] proposed two service selection approaches which are distributed and trust aware. First approach is based on Bayesian networks. Second approach is based on beta-mixture model. Authors validated their approach simulation implementation. They claim both approaches accurately punish and reward services through results in terms of the qualities.

S. Chibani et al [18] presented a meta-heuristic bio-inspired approach which addresses the QoS aware web service composition. This work is based algorithm called Elephant Herding Optimization. The presented algorithm is inspired by one of the behaviours of elephant group which is herding. The characterization is done by a process of dividing and combining the population which produce sub populations called clan. The work by author allows local searches to move toward a global optimum in order to exchange information needed for optimization.

H. Ayed et. al. [19] proposed an enhanced algorithm based on ant colony optimization. An approach is based on a swap concept for the QoS which addresses web service selection problem. The main aim of work is avoiding the trap of local optima along with reduction in the search space and time. In this work authors also claim better solution in terms of quality which avoids stagnation.

In [20], A. Tripathy presented automatic service selection framework for composition of web service-based systems. An automatic monitoring at run-time of system is facilitated by framework presented by authors. An automatic re-composition strategy is adapted by monitoring for discrepancy cases if any.

In [21], R. Liu introduced analysis and discovery of the distribution of customer requests which is used to identify request clusters. Authors also performed service patterns mining from historical service solutions. In order to improve the efficiency of service selection algorithm, probability matrix is generated. The request-solution mapping between request clusters and service patterns is also generated which contains statistical analysis.

In [22] Y. Zang et. al defined and explained the long/short-term utility along with their preferences provider, consumer, and operator as parties. The work also established the model with consumer's short-term utility as well as long-term utility of providers at same time. To perform estimation of long-term utility of accepted task, service providers are considered. This work also proposes Non-dominated sorting Genetic Algorithm for solving optimization problem in service composition. The proposed algorithm uses and combines tabu search and k-means mechanism to get set of optimal solution.

[5] W. Ahmed et. al. proposed Hidden Markov Model for QoS metrification. This method helps in finding path for the execution of requests. The response time and cost QoS of web service is used to measure and predict behaviour. The quantity of QoS attribute is used to rank services instead of only quality parameters. The observation on feasibility and usefulness is done through implementation of model. Authors claims automatic selection the best and reliable web service and with good prediction accuracy.

[23] X. Liang et. al. introduced concept of internal complementarity. Using this concept authors addressed to solve the problem of QoS-aware web service selection. In this work selection problem is mapped to a multi-choice multi-dimensional knapsack problem. Also, they performed and demonstrated complexity analysis of some existing approaches related to selection problem. This work proposes a framework for deriving the solution iteratively by considering solution structure and QoS constraints. Some of the heuristic approaches are implemented for the verification of framework in terms of effectiveness.

Q. Yu et. al. [24] proposed composite service skyline with dominance relationship to find best candidate services which will reduce search space for all possible combination's solution. This work also proposed bottom-up framework for service computation using skyline to scale up with large number of candidate services in composition. Authors claim on evaluation of scalability, effectiveness, and efficiency of composite service.

In the above surveyed works, most of the work considers QoS for service computation with few of them addressed service correlation between candidate services. But no work is address correlation of candidate service with QoS constraints for pruning candidate services for selection to compose composition.

3. PROBLEM DEFINITION

Definition of the problem is given in this section. The definition of web, service, web service, service execution plan and service composition are described. QoS correlation is given to formalize proposed approach.

Definition 3.1 (Web). Web is collection Hyper Text Markup Language (HTML) or advanced html pages kept on servers (like apache tomcat, BEA weblogic, IIS server etc.) and connected to internet is also known as World Wide Web (WWW).

Definition 3.2 (Service). The functioning required serving task or activity belongs to any physical work. Example, Airline services, Hotel services, Train services, Hospital services, Television services and Banking services etc.

Definition 3.3 (Web Service). A Web service is a tuple with four parameter $S_w = (UID, DESC, I, QoS)$, where: (1) UID is the primary identification number of the web service; (2) $DESC$ is description of the service's provided functionalities, with the input required, output received, pre-condition required and visible result of the web service; (3) I is the mandatory information of the web service, with its full name on WWW, URL location, and service provider. (4) QoS is a set of quality parameters, with response time, price, availability, reputation, etc.

In Definition 3.3 of web service, each web service is uniquely identified by UID , $DESC$ informs about the functions available with a web service, which is used to aggregate candidate services for tasks in the service plan, and QoS will be defined in detail by Definition 3.6.

Definition 3.4 (Service Execution Plan). A service execution plan is a triple $SEP = (ST, EP, SI)$, where: ST is a set of tasks, EP is execution probability of plans, SI is the structural information of execution plan. A service execution plan is description of a business process. There may be lots of web services with similar functions and varying QoS and providers to achieve almost all tasks.

Definition 3.5 (Composite Web Service). A composite web service is a Triple $CWS = (WS, SI, QoS)$, where: WS : the composite web service is composition of different Web services (WS); SI gives information about structure of the web service plan. QoS is quality of service parameter of composite web service. Composite web services are created by selection process of on web service among multiple web services with different QoS and same functional description for each task which is provided according to service execution plan's structure.

In the process of service composition, a service execution plan is specified initially by the user. Then, by keeping in mind structure of service execution plan, process of service selection is carried out to select web component services. At the last, composition of the selected web services is performed. In the process of composition, many steps are involved; out of all service selection is considered as most important step. Because this step decides the composite service's quality. The symbol " \parallel " is used to represent the sequential flow of composition between two services. Example, $ws(p) \parallel ws(q)$ represents a composition of web services is composed of $ws(p)$ and $ws(q)$ in sequential manner.

In Real-time some QoS of web service are depending correlated to other services' QoS attribute. Quality attributes of a web service may have correlation with two or more service's quality attributes and may have two or more service whose QoS attributes have correlation with more than one service's quality attributes.

Definition 3.6 (QoS). QoS is Quality attribute of web service and QoS_i is attribute set with a four-tuple, $QoS_i = (IV, CV, TO, FROM)$, where: (1) IV is the initial value of QoS_i of the web service; (2) CV is the value with correlation of QoS_i of the web service; (3) TO is the set of services that the value of QoS_i is correlated to; (4) $FROM$ is the set of services whose value of QoS_i is correlated from set of TO .

Each service may have several QoS attributes, and each QoS attribute is defined by four elements. The default value applies to the service if no services in $S1$ are invoked first, while correlated value applies to the service if and only if one or more services in $S1$ are invoked first. If one service is selected, then all the services in its $S2$ will appear with their correlated value.

Definition 3.7 (Correlation). A correlation is process of inferring a relationship with one attribute is correlated to another. This is represented as a triple $Qc = (QoS_i, TO, FROM)$, here the QoS attribute QoS_i of $FROM$ is correlated to TO .

4. PROPOSED APPROACH

In this section, we present our correlated service selection model as shown in figure 1. It consists of four modules.

1. Task Management Module: The purpose of task management module is to assign component service to task. This can be done using service's functional attributes. Example, Credit card service of bank, Hotel booking service, route service of map etc.

2. QoS Management Module: This module is used to manage uncertain values of QoS of each component service. QoS value can be derived from service providers (i. e. cost and availability).

3. QoS Correlation Module: This module allows to store the correlated values of QoS of each service. The correlation value of QoS service can be derived from QoS repository. Example, correlated value of QoS cost attribute (value 20) of service A is cost 15.

4. Service Selection Module: It provide dominants service composition for service request from each user. This part is consisting of algorithms.

I. Service pruning algorithm which is used to reduce search space of number of candidate services.

II. Second algorithm is used to rank services and give final selection with correlated values of QoS of services.

The service ranking algorithm sorts the candidate services of each task. The dominance relationship and its some variants are also used in [4, 7, 8, and 24]. The skyline-based service ranking is used in [25-29].

4.1 Service Pruning

The best optimal solution form combination of services gets by selecting services from complete set of candidate services without pruning process [30, 31]. When the number of candidate services is very large in number then exponential time is required to get solution of service selection problem. For example, a business process with n tasks, m candidate services per task, the possible number of solution forms with given candidate services is mn . But feasible solution does not contain all correlated candidate services available per task. To deal with this situation, we propose service pruning on correlated candidate services. The pruning process reduce the number correlated candidate services per task and hence number of possible combinations of feasible solution is also reduce while guarantee of finding optimal solution. In short pruning process improves the efficiency of service selection process by search space reduction.

4.2 QoS Constraint-based Pruning

The goal is to keep only candidate services which satisfy the global constraints and remove those one which violate the global constraints. In [32] given some global constraints-based methods for removing such candidate services. For example, if the user requires that the global response time of the composite service should not exceed 5 units of time, any service with a response time greater or equal to 5 units of time can be eliminated. The QoS constraint-based pruning process select satisfactory service with guarantee of optimal solution. It is depending on local as well as global thresholds of candidate service. Also, satisfaction of global constraints is decided by computation of local QoS of candidate service's threshold. In composite service, if any one candidate service does not satisfy local threshold then it composition will violate global constraints. In such case, candidate service can be removed from set of candidate services of task which will reduce the search space.

Let consider process with p sequential tasks i.e. T_1, T_2, \dots, T_p with $Q(\text{Response time})=10.0$ units time and $Q(\text{cost})=5.0$ units cost as two global QoS constraints. In the computation of local thresholds of QoS for all p tasks, we need to identify lower bound of QoS values of both response time and cost for all p tasks. In our case $C(T_1, \text{Response time})_{\min} = 1.0$, $C(T_2; \text{response time})_{\min} = 1.0$, $C(T_3; \text{response time})_{\min} = 1.0$ and $C(T_1, \text{Cost})_{\min} = 1.0$, $C(T_2; \text{Cost})_{\min} = 1.0$, $C(T_3; \text{Cost})_{\min} = 1.0$ for $p=3$. Now, we compute local QoS thresholds for each quality attribute for all three tasks. For instance, as the response time is an additive attribute and the minimum response time value of the task T_1 is equal to 1.0 in the best case, all services with a cost greater than 9; $(C(\text{response time}) - C(T_1, \text{response time})_{\min}) = 4$ of the task T_1 cannot be part of a feasible solution. The composed service will violate

global constraints if any one QoS attribute's value of participating candidate service does not satisfy condition of a local threshold then it can be removed from candidate service set of tasks. In our proposed approach only candidate services that violate the global constraints will be discarded from the set of candidate services of tasks.

A value of local threshold $C_{LTH}(T_i, q_p)$ for the p th attribute of the task T_i depends on the value of the required global constraint $C(q_y)$ and the minimum and maximum values of this QoS attribute that can be offered by services (i.e., $C(T_i; q_p)_{min}$ and $C(T_i; q_p)_{max}$).

For this, we propose to compute local thresholds for each business task based on a constraint optimization model. This model can be applied for generic business process structures that can contain sequential, parallel, choice and loop patterns with several categories of quality attributes. For simplicity, the proposed model assumes that business processes contain only atomic activities.

The proposed model is as follows:

$$f(q_p) = \text{Minimize} \sum_{T_i=1 \text{ and } i \neq j}^{i=p} C(T_i, q_p) - C(T_j, q_p) \quad \dots \quad \text{Eq. (1)}$$

The QoS constraint-based pruning algorithm is presented and explained in this section. These steps are as in algorithm 1. The input to algorithm is the set of candidate services (CS_i) without any quality constraints. The output is set of candidate services after pruning process which is reduced set of services. Line 1 sets the set of pruning candidate services to null. Line 2 sets the service threshold to 0; Line 3 gets minimum value i_{th} quality attribute. Line 3 gets maximum value i_{th} quality attribute. Line 5 sets the value of Q to Response time and Cost. Line 6 retrieves the first (i.e. Response time) from set of Q . A for loop with empty condition checking is written in line 7. Line 8 adds local thresholds value to all tasks for both quality attribute i.e. Response time and cost. Line 9 gets next quality attribute for threshold computation from set of Q . Line 10 and 11 add current j_{th} candidate service of i_{th} task and check it for empty condition. Line 12, 13, and 14 checks each service is satisfying thresholds value or not. If constraint not satisfied by service will get added to CS_i . If constraint is satisfied by service, it will be added in PCS_i as in line 16. Line 17 and 18 indicates if PCS_i is empty means no service violates constraint set by threshold.

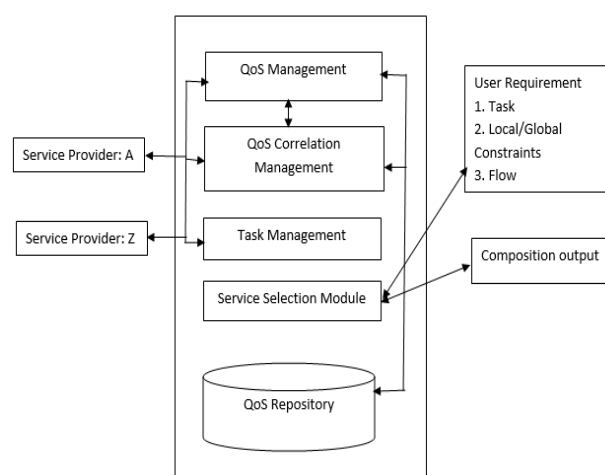


Figure. 1 Service Selection Framework.

Algorithm 1: *Pruning of candidate services for Task T_i*

Input: *The set of candidate services CS_i*

Output: *The set of pruned candidate services PCS_i*

```

1       $PCS_i = \text{null};$ 
2       $C_{LTH} = 0;$ 
3       $Min = \text{minValues}(CS_i);$ 
4       $Max = \text{maxvalues}(CS_i);$ 
5       $Q \leftarrow \{\text{Response time, Cost}\};$ 
6       $q_p \leftarrow \text{first}(Q);$ 
7      For  $q_p$  not empty do
8           $C_{LTH}(T_i, q_p) \leftarrow \text{localThreshold}(T_i, q_p)$ 
9           $q_p \leftarrow \text{next}(Q);$ 
10          $PCS_{ij} \leftarrow \text{first}(CS_{ij})$ 
11         For  $PCS_{ij}$  not empty do
12             For each  $q_p$  in  $Q$  do
13                 If  $C(CS_{ij}, q_p) > C_{LTH}(T_i, Q_p)$  Then
14                      $CS_i = CS_i \text{ Union } CS_{ij}$ 
15                     Break;
16                  $PCS_i = PCS_i \text{ Union } CS_{ij}$ 
17                 If  $PCS_i = \text{null}$ 
18                     Then no pruning.
```

Algorithm 2: *Candidate service selection for task T_i*

Input *Set of pruned candidate services CS_i*

output *Selected candidate service S_i*

```

1      For each  $S_{ij}$  from set of  $CS_i$  (set of pruned service)
2          Compute utility of service  $UF(S_{ij})$  using equation 2.
3          Do ranking of service using utility values obtained from eq.2.
4          For each  $S_{ij}$  from set of  $CS_i$ 
5              For each  $q_p$  in  $Q$ 
6                  Sort services by ascending order
7                  Select  $S_{ij}$  with minimum Normalized Quality value.
8                  Break;
```

First, we compute utility of each candidate service using utility function UF as given in equation 2 (lines 3 and 4). The utility of each candidate service is nothing but normalized QoS values of each service computed. The Q_{max} and Q_{min} in computation of normalized values helps in avoiding local optima.

$$UF(S_{ij}) = \sum_{i=0}^p W_i * \frac{Q_{max} - Q(S_{ij})}{Q_{max} - Q_{min}} \quad \text{---Eq.(2)}$$

Describe ranking here. Perform ranking using normalized QoS value of service using equation 2. Line 5 explain ranking of services based on normalized values which we computed using sorting (line 6 to 9) of normalized QoS values by ascending order of each candidate services. At last line 9 select the candidate service with minimum normalized QoS value.

5. EXPERIMENTATION

Experimental simulation is performed on a system with a 64-bit Intel Core i3, 4GB RAM and Fedora 20 operating system. To implement the proposed algorithms, we used Java 1.8 programming language. The generation of candidate services and correlation between them for each abstract task were performed randomly. The correlation percentage is 50% i.e. for every alternate candidate service we created correlated service with QoS offering to half the value of original values. The response time and cost QoS attributes are taken with their minimum and maximum values using poison distribution. The response time constraint and cost constraint were randomly generated. The response time and cost are generated assuming a poison uniform distribution over the interval [1.0, 5.0] and [1.0, 10.0] respectively. The composite services considered in the implementation work have a sequential structure. The computation time (selection time) is noted by varying the number of composite services per task, and the number of tasks per activity. The weight of both QoS attributes i.e. response time and cost is fixed to 0.5 under the condition that the sum of all the weight values should be equal to 1.

6. RESULTS AND DISCUSSION

The computation time is used to evaluate the performance of the proposed algorithms. The computation time represents the selection time of the proposed algorithm. **Computation Time versus Number of Services:** The simulation is validated for the scalability of the algorithm. The number of tasks is set to 3 initially and the number of candidate services for task varies from 100 to 1000. As shown in Fig. 2, the computation time increases with the number of candidate services satisfying the QoS threshold values increases. Then number of tasks is set to 5 and the number of candidate services for task varies from 100 to 1000. As shown in Fig. 3, the computation time increases with the number of candidate services satisfying the QoS threshold values increases. Fig. 2 and 3 both shows that approach without quality constraint with correlation method is taking more computation time for selection whereas our approach is taking less computation time except some cases i.e. number of candidate services per task is 900 is taking less time for old approach for task=3 and number of candidate services per task is 800 is taking less time for old approach for task=5. Possible reason behind requirement of less computation time maybe it contains a greater number of quality constraints and pruned services may be more in it.

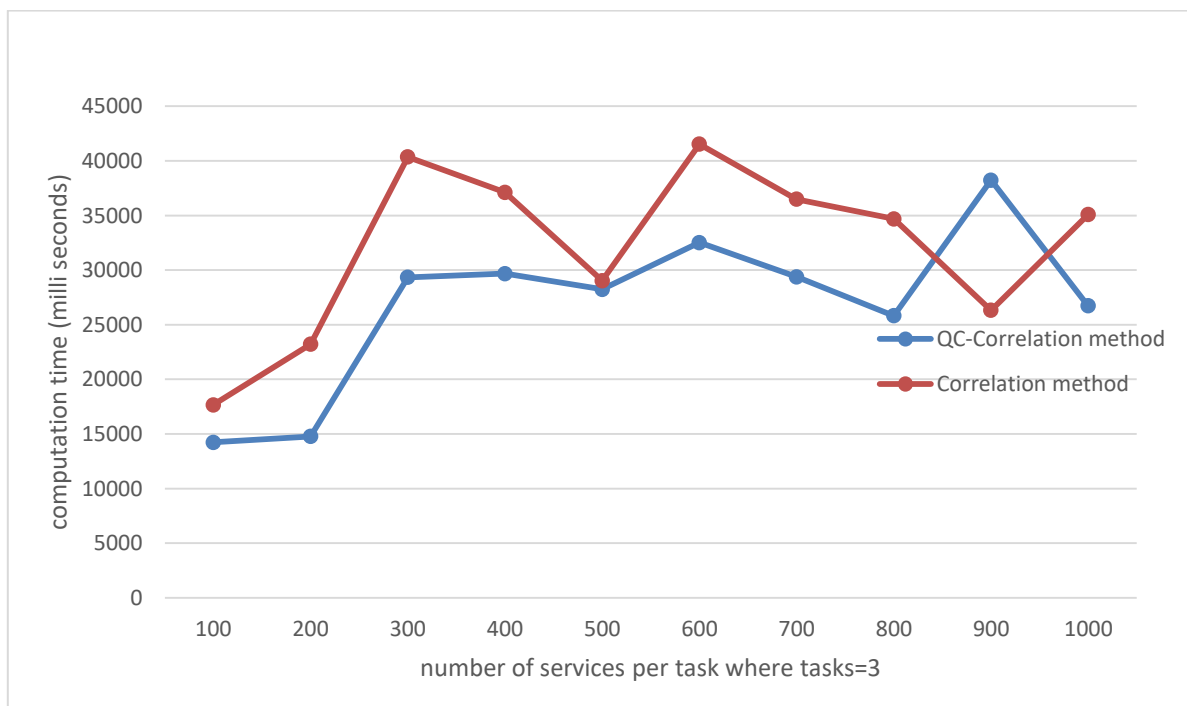


Fig.2: The comparison of proposed algorithm with existing approach for tasks=3

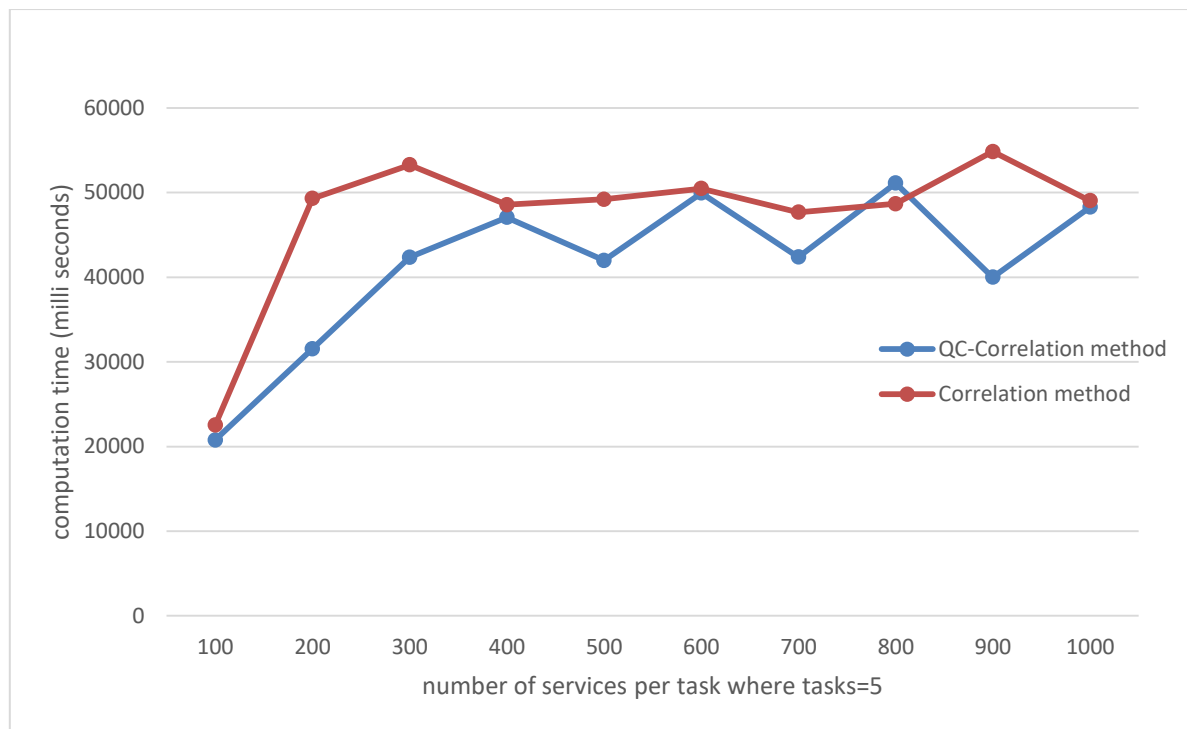


Fig.3: The comparison of proposed algorithm with existing approach for tasks=5

7. CONCLUSIONS

In this paper, we have proposed approach which allows handling of quality constraints of candidate services and QoS correlation between candidate services. The proposed service selection approach is based on constraint optimization model which allows selection of best combination of candidate services to compose composite service. From experimental simulation results it is presented that approach with quality constraint along with services correlation outperforms existing method. The results show that proposed method is scalable enough for 1000 number of services per task which indicate large environment.

However, in this paper sequential workflow is considered for making plans. Hence we will consider our future work to other workflow structure.

References

- [1] G. Alonso, F. Casati, and H. Kuno, "Book on Web Services: Concepts, Architectures and Applications", Springer, (2010), pp. 1-369.
- [2] R. Micillo, S. Venticinque, N. Mazzocca, and R. Aversa, "An Agent-Based Approach for Distributed Execution of Composite Web Services," IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, (2008), pp.1-6.
- [3] Y. Liu, A. Ngu, L. Zeng, "Qos Computation and Policing in Dynamic Web Service Selection", Proceedings of 13th International Conference on World Wide Web, (2004), pp. 66-73.
- [4] V. Cardellini, E. Casalicchio, V. Grassi, F. Presti, "Flow-Based Service Selection for Web Service Composition Supporting Multiple QoS Classes, (2007), pp. 743-750.
- [5] A. Masri and Q. Mohmoud, "QoS based discovery and Ranking of Web Services." Proceedings of IEEE conference on computer comm and networks, (2007) pp. 529-534.
- [6] W. Ahmed, Y. Wu, W. Zheng, "Response Time based Optimal Web Service Selection", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, (2013), pp. 1-11.

- [7] F. Wang, Y. Laili and L. Zhang, "A many-objective memetic algorithm for correlation-aware service composition in cloud manufacturing," *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, vol x, no. x, (2020), pp. 1-20.
- [8] D. Li, D. Ye, N. Gao and S. Wang, "Service Selection With QoS Correlations in Distributed Service-Based Systems," in *IEEE Access*, vol. 7, (2019), pp. 88718-88732.
- [9] S. Deng, H. Wu, D. Hu, and J. Zhao, "Service Selection for Composition with QoS Correlations", *IEEE TRANSACTIONS ON SERVICES COMPUTING*, vol. x, no.x, (2014), pp-1-14.
- [10] Y. Du, H. Hu, W. Song, J. Ding and J. Lu, "Efficient Computing Composite Service Skyline with QoS Correlations", 15th IEEE International Conference on Services Computing, 2015, pp.41-48
- [11] L. Purohit and S. Kumar, "A Classification Based Web Service Selection Approach," in *IEEE Transactions on Services Computing*, doi: 10.1109/TSC.2018.2805352, (2018), pp.1-14.
- [12] W. Wang, Z. Huang and L. Wang, "ISAT: An intelligent Web service selection approach for improving reliability via two-phase decisions," *Journal of Information Sciences*, vol 433, no 434, (2018), pp. 255-273.
- [13] R. Z. Yasmina, H. Fethallah, and D. Fedoua, "Selecting Web Service Compositions Under Uncertain QoS," Springer International Publishing, CIIA 2018, IFIP AICT 522, (2018), pp. 622-634.
- [14] S. Kurdija, M. Silic, G. Delac and K. Vladimir, "Fast Multi-Criteria Service Selection for Multi-User Composite Applications," in *IEEE Transactions on Services Computing*, vol. x, no. x, (2019), pp. 1-14.
- [15] M. Moghaddam and J.G. Davis, "Simultaneous service selection for multiple composite service requests: A combinatorial auction approach," *Decision Support Systems*, (2019).
- [16] M. E. Khanouche, N. Atmani and A. Cherifi, "Improved Teaching Learning-Based QoS-Aware Services Composition for Internet of Things," in *IEEE Systems Journal*, vol. 14, no. 3, (2020), pp. 4155-4164.
- [17] C. HANG and M. SINGH. "Trustworthy Service Selection and Composition" *ACM Trans on Autonomous and Adaptive Sys.*, Vol. 5, No. 4, (2010), pp. 1-18.
- [18] S. Chibani, and A. Tari. "Elephant Herding Optimization for Service Selection in QoS-Aware Web Service Composition", *International Journal of Computer and Information Engineering*, Vol:11, No:10, (2017).
- [19] H. Ayed, F. Dahan, T. Alfakih, H. Mathkour, and M. Arafah, "Enhancement of Ant Colony Optimization for QoS-Aware Web Service Selection" *IEEE Access*, (2019), pp.1-12.
- [20] A.Tripathy, M. Patra, S. Pradhan, R. Raque, P. Mohanty, and S. Dash, "Graph Based Service Selection for Composition and Adaptation" *The Research Bulletin of Jordan ACM*, Volume II(II), (2011), pp. 83-89.
- [21] R. Liu, X. Xu, Z. Wang, and H. Xu, "Probability Matrix of Request-Solution Mapping for Efficient Service Selection", In the proceedings of IEEE 24th International Conference on Web Services, (2017), Pp. 444-451.
- [22] Y. Zhang, F. Tao, Y. Liu, P. Zhang, Y. Cheng, and Y. Zuo, "Long/short-term utility aware optimal selection of manufacturing service composition towards Industrial Internet platform", *IEEE Transactions on Industrial Informatics*, (2019), pp. 1-11.
- [23] X. Liang, A. Qin, K. Tang, K.Tan, "QoS-aware Web Service Selection with Internal Complementarity", *IEEE Transactions on Services Computing*, (2016), pp-14, vol. x, no. x.
- [24] Q. Yu and A. Bouguettaya. "Efficient Service Skyline Computation for Composite Service Selection". *IEEE Transactions on Knowledge and Data Engineering*, (2011), 268- 281.
- [25] L. Purohit and S Kumar, "Exploring K-Means Clustering and skyline for Web Service Selection", 11th International Conference on Industrial and Information Systems, (2016), pp. 603-607.
- [26] L. Purohit and S Kumar, "Clustering based Approach for Web Service Selection using Skyline Computations", *IEEE International Conference on Web Services*, (2019), pp. 260-264.
- [27] Y. Yang, F. Dong, J. Luo, "Computing Service Skycube for Web Service Selection", *Proceedings of IEEE 19th International Conference on Computer Supported Cooperative Work in Design*, (2015), pp.614-619.
- [28] M. Moradi, and S. Emadi, "A Review of Service Skylline Algorithms", *Journal of Soft Computing and Decision Support Systems*, Vol. 3, No. 3, (2016), pp.55-58.
- [29] A. Ouadah, A. Hadjali, F. Nader, and K. Benouaret, "SEFAP: an efficient approach for ranking skyline web services", *Springer-Journal of ambient Intelligence and Humanized Computing*, Vol. x, no. x, (2018), pp.1-16.
- [30] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. "Qos-aware middleware for web services composition." *IEEE Transactions on Software Engineering.*, vol. 30, no. 5, (2004), pp. 311-327.
- [31] A. Hassine, S. Matsubara, and T. Ishida. "A constraint-based approach to horizontal web service composition." *Proceedings of International Semantic Web Conference*, (2006), pp. 130-143.
- [32] Q. Yu and A. Bouguettaya. "Computing service skyline from uncertain qows". *IEEE Transactions on Services Computing*, vol.3,no.1,(2010),pp.16-26.