I.J. Information Technology and Computer Science, **2019, 4, 44-53**
Published Online April 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2019.04.05

Subset Matching based Selection and Ranking (SMSR) of Web Services

[1]Md. Abdur Rahman, [2]Md. Belal Hossain, [3]Md. Sharifur Rahman, [4]Saeed Siddik
[1]Centre for Advanced Research in Sciences, [2,3,4]Institute of Information Technology
University of Dhaka, Dhaka, Bangladesh
E-mail: {[1]mukul.arahman, [2]belalkucse, [3]sharif.lalon}@gmail.com, [4]saeed.siddik@iit.du.ac.bd

Received: 06 January 2019; Accepted: 20 January 2019; Published: 08 April 2019

Abstract**—**Web service is a software application, which is accessible using platform independent and language neutral web protocols. However, selecting the most relevant services became one of the vital challenges. Quality of services plays very important role in web service selection, as it determines the quality and usability of a service, including its non-functional properties such as scalability, accessibility, integrity, efficiency, etc. When agent application send request with a set of quality attributes, it becomes challenging to find out the best service for satisfying maximum quality requirements. Among the existing approaches, the single value decomposition technique is popular one; however, it suffers for computational complexity. To overcome this limitation, this paper proposed a subset matching based web service selection and ranking by considering the quality of service attributes. This proposed method creates a quality-web matrix to store available web services and associated quality of service attributes. Then, matrix subsets are created using web service repository and requested quality attributes. Finally, web services are efficiently selected and ranked based on calculated weights of corresponding web services to reduce composition time. Experimental results showed that proposed method performs more efficient and scalable than existing several techniques such as single value decomposition.

Index Terms**—**Web service composition, selection, ranking, SVD, subset.

I. INTRODUCTION

Web service is a technology by which two or more remote web applications interact with each other over network [1]. Software applications are usually written in several programming languages and running on various platforms, but those applications internally use web services to exchange data over computer networks like the Internet. This interoperability between programs (e.g. PHP and Python) or platforms (e.g. Linux and Windows) is provided with several open standards named as XML, SOAP, HTTP, etc. However, single web service cannot satisfy entire user requirements, because requirements often need multiple service providers. To deal with this, researchers have begun to develop different types of web service composition approach to achieve the original reality of web services. Composition of individual web services to obtain a web process is called web service composition [2]. This composition is useful when the user request for a service with specific input-output, which cannot satisfy by single web service [3]. In this approach, Quality of Service (QoS) becomes a significant factor for selecting effective services among the different available web services [4]. That is why, QoS is the most important differentiating point for a set of web services which provides similar types of functions [5].

Several web service composition methods have been proposed to provide best services to the user which can be categorized into spatio-temporal based [2], context based [3], semantic based [4, 6], agent based [7], QoS attributes based [8], etc. Using spatio temporal information Neiat et al. proposed web service selection and composition approach for crowd sourced services considering new QoS criteria [2]. In this technique variation of Dijkstra shortest path finding algorithm is used to minimize the service search cost. Context and policy based composition technique to manage web service behaviors exposed during composition is proposed by [3]. The system comprises with policy, user, web service and resource layers. Automatic service registry and discovery based composition strategy has been proposed to provide integration and composition using semantic web service integration life cycle [4]. Target oriented, interactive composition approach was proposed to filter and select semantic web services for web service composition [6]. Software agent and context oriented web service composition approach has also been used to reduce web service composition complexity [7]. For selecting and ranking of most relevant web services, chan et al. proposed single value decomposition based web service selection and composition method [8]. A Quality of Services matrix (QoS matrix) and QoS Web service repository (QW repository) are used on their technique. The quality attributes and closely related web services are placed sequentially, where relevant services are recommended. However, existing web service selection and composition methods based on singular value decomposition suffer for calculation complexity, specifically, when the QoS matrix size increases. In addition, when the QoS attribute set does not match completely with repository web services, the new request attributes are excluded and a new set is also created using other attributes, which lead to performance degradation.

Therefore, a Subset Matching based Selection and Ranking (SMSR) technique is presented in this paper to select and rank most relevant web service with optimal response time and accuracy. The proposed SMSR framework automatically selects and composes web services using QoS attributes as shown briefly in “Fig. 1”. When an agent program sends request with QoS attributes to get web services, the framework first creates a QW matrix to store available web services and associated QoS attributes. Then, this framework creates another subset.

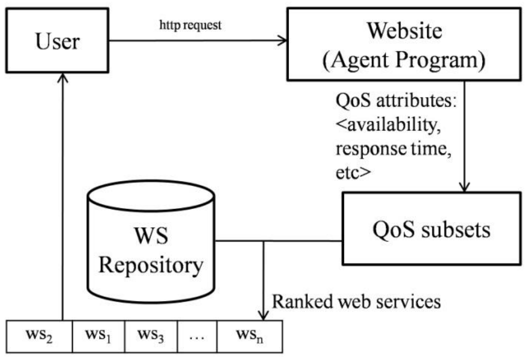

Fig.1. Short overview of SMSR

Selection and ranking process are performed based on matching weight of individual web services which is obtained from a subset QW matrix. The whole process has been implemented in four steps named as forming QW matrix, generating subset from QoS attributes, creating subset QW matrix, and ranking procedure.

The proposed web service selecting and ranking framework has been evaluated with time complexity and accuracy key factors. This method has been experimented on pre-selected 10, 20, 30, 40 & 50 number of web services. The average response time and accuracy were reported 0.56s and 0.81 respectively. Where, the existing single value decomposition technique performs 0.78s and 0.71 in average response time and accuracy respectively. It is found that, the response time of proposed method is 39% better than exiting one. The investigated results reveal that the use of subset matching approach proves the efficiency of web service selection and ranking, in terms of response time and accuracy.

The rest of this paper is organized as follows. The Section II describes the existing related works and problem analysis. The proposed approach will be discussed elaborately in Section III. Then comparative result analysis will be illustrated in Section IV. Finally, conclusion with future work is given.

II. RELATED WORKS

Web service is a standardized way of communication between heterogeneous software applications over HTTP using technologies such as XML, SOAP, WSDL, and UDDI.As uses of web services increasing day by day, researchers have investigated various web service selection and composition approaches which can be divided into different groups such as spatio-temporal based [2], context based [3], semantic based [4, 6], agent based [7], etc. Several web service composition techniques are discussed in this section.

Neiat et al. proposed a spatio-temporal based selection and composition approach for crowd sourced services considering new QoS criteria [2]. In this technique variation of Dijkstra shortest path finding algorithm is used to minimize the service search cost. Using this algorithm, a directed graph is generated where each vertex represents with web service which has associated space-time-attribute and edges have QoS attributes. The services are indexed using the spatiotemporal features in a 3D R-tree. Using location and time, compossable services are searched and confirm that coverage of composition not changes in space and time. The effectiveness of proposed filtering method is experimented in terms of composition time. The result shows that proposed strategy considerably reduces the computation time and a little optimality ratio increases with the number of services. However, hot spot web services with static in nature assumption have been considered for experimentation which leads to static composition plan.

Maamar et al. investigated context and policy based composition technique to manage web service behaviors during environment changing [3]. This system describes how to make web services responsive in case of any changes in the environment and how to solve the problem of automatic participation of web services in composition. It also considered either delay or reject in the participation of web services, to control the overloaded web services. It comprises of four layer named as Policy, User, Web Service and Resource. Where, policy layer defines how the web service will react based on composition progress. User layer represents the users who are looking for web services. On the other hand, web services that are advertised on various registries are represented by the web service layer. Finally, resource layer represents the computing means, upon which the web service will operate. When user submits request to composition manager, component manager discovers the required services from UDDL. The list of web services those are selected by component manager have been sent to policy manager, which makes web services bind to appropriate behavior. At behavior binding time, different contextual information is considered, which were received from context manager. The environmental information is generated by context manager considering users, resources, web services and other policies. This process incorporates context and policy together, but, web services identification resources and user’s preference record are required to store in memory for web service composition, which is very difficult for huge number of resources and dynamic nature of users. If the resources are infinite, the problem of this solution will be

NP-hard problem because it has to keep in memory all the resource identification record.

Aslam et al. implemented automatic service registry and discovery based composition strategy to provide semantic web service integration and composition using semantic integration life cycle [4]. This life cycle consists of multiple modules named as business process modeling, development, semantic workflow enrichment, runtime phase, and service management. The cycle follows top down execution approach starting from business process to their execution. The business process modeling defines business logics, technique of single process combination with others data flow between these processes and control. Where, development phase defines technical description of business process. On the other hand, semantic service requests are annotated and business processes are defined in semantic phase. In runtime phase, semantically enriched workflows are deployed on semantic enable execution engines. At the end in service management phase, all the services are managed and alive through the lifecycle. It has been reported that the complexity and time of web service composition are less than existing solution. This process works on WSDL and does not support the specification of various constraints such as management statement, service level agreement and other constraints. To support these issues another system needs to be incorporated which is unavailable in every case.

Sirin et al. presented target oriented, interactive composition approach to filter and select semantic services for web service composition [6]. For interactive composition match-making algorithm is used to filter web services at the service composition time. It gradually generates the composition with a forward and backward chaining of services. At each step of composition, this system adds a new service to the composition, and filters further possibilities based on the current text and user decision.

A software agent and context oriented web service composition approach has been proposed to reduce service composition complexity [7]. It acts as user negotiates with their peers to agree on the web services among the participating process during the composition. An agent is context aware and can deal different complex issues regarding the web service composition such as business capacity of provisional web services, time of web services occurs, etc. Three types of agents named as composite agents for composite services, master agents for web services and service agents for web service instances are addressed over the research. The agents are aware of I-context, W-context and C-context which facilitate the proposed system.

Chan et al. proposed single value decomposition based web service selection and composition method in order to recommend most closely related services to users [8]. A quality of services matrix (QoS matrix) and web service repository (QW repository) are used in this technique. Where, QoS matrix and QW repository contains quality of services parameter and patterns of web services respectively. A QW matrix has been generated from the transformation of initial QoS attributes and web services set obtained from the QW repository. After completing the generation, it is decomposed by SVD transformation to an *n*-dimensional QW space. Thus, QoS attributes and closely associated web services are placed near one another. Finally, the most closely related service is recommended to users, which can be either accepted or rejected. However, this SVD based Web service selection and composition method suffers for calculation complexity specifically when the QoS matrix size increases. In addition, when the QoS attribute set does not match completely with repository web services, the new request attributes are excluded and a new set is created by using other attribute. Then a pseudo web service is selected to predict recommend web service for the new attributes. But this recommended web service may not be related to the required attributes which degrade web service selection and composition performance.

Kwon et. al. investigated redundancy free web service composition using two phase algorithm named forward and backward phases [9]. The candidate web services which participate in composition are selected for searching an index linked in forward phase. Linked index is built over web services according to their connectivity. By deleting unnecessary services, redundant free web services composition from candidate web services are generated in backward phase. Almost 10K synthetic web services are used for experimental analysis. Results showed that proposed technique performs better than other methods.

Lamparter et al. proposed preference based highly configurable web service selection method using utility function policies, where optimal service selection framework combining logical rules with optimization [10]. Ontology concept is used to define service attributes with corresponding values semantically. It includes offers and requests, which leads to define appropriate attributes value. An optimal service selection mechanism and flexible matching algorithm are the major contribution of this research. The investigational results showed that proposed algorithm produce optimal result with 2sec overhead compare to random service selection method. However, additive preference functions are considered for the experimentation.

Hachemi et al implemented user preferences and case-based planning method to improve web service compositions' effectiveness using previous successful experiences [11]. This approach uses case-based reasoning and planning artificial intelligence techniques. The whole process is implemented into seven phases which are service request, translation, retrieval, planning, adaption, execution and learning. User preferences are integrated at selection, adaptation and planning phases. For a user request the system generates the by finding a composition plan from the library of cases. The new user request and solution plan are stored as a case to reuse in future. For implementation, Java programming language, PDDL3.0 and SGPlan are used.

The analysis of existing composition approaches showed that different strategies have been proposed for web service selection and composition such as agent based, semantic based, context based etc. Using QoS parameter with SVD is introduced to select and ranking web services where computational complexity increases with the increases of QoS attributes. Therefore, mitigation of their drawbacks as well as other mentioned problems are essential in order to select and rank top most relevant web services for users. However, there is no appropriate guideline to select and rank web services with QoS attributes, which increase the composition efficiency in terms of selecting and ranking.

III. Proposed Method

Subset Matching based Selection and Ranking (SMSR) of web services framework have been proposed for automatic selection and composition. SMSR method creates **Q**uality of service and **W**eb services (QW) matrix to store available web services and supported QoS attributes. When an agent website sends request with QoS attributes, the system creates another subset of QW matrix using submitted attributes and entire QW matrix. Selection and ranking process is performed based on the QW matrix subset. The whole process is divided into following four steps and demonstrated in "Fig.2".

Step 1: Forming QW Matrix
Step 2: Generating Subset from QoS Attributes
Step 3: Creating Subset of QW Matrix
Step 4: Selecting and Ranking Procedure

Detail descriptions of the above steps are given elaborately in the following sub-sections.

A. Step 1: Forming QW Matrix

Web services with their associated QoS attributes are arranged into QW matrix which is the foundation of this proposed SMSR framework. In QW matrix, QoS attributes are entered as rows and the web services are entered as columns. *N* number of web services and *M* number of QoS attributes are arranged in *N* × *M* rectangular matrix considering that every web service is supported by one or more repository QoS attributes as shown in Table 1.

Table 1. QW Matrix

Attributes	W_1	W_2	W_3	...	W_n
x_1	*1*	*1*	*0*	...	*0*
x_2	*1*	*0*	*0*	...	*1*
x_3	*0*	*0*	*0*	...	*0*
...	..	...	...	...	*0*
x_m	*0*	*0*	*1*	...	*0*

In Table 1, *W* represents web service providers, where W_1, W_2, W_3, and W_n indicate provider-1, provider-2, provider-3 and provider-*n* respectively. And *x* is used for QoS attributes for example x_1, x_2, x_3, x_n for price, availability, security and *n*-attribute respectively. Every entry in QW matrix is called a Quality Index (QI) for a pair of QoS attribute and web service. In QW matrix, QI value indicates the available or unavailable of a QoS attributes for the corresponding web service. Where value *1* and *0* indicates availability and unavailability of QoS by particular web service provider respectively.

QI value of QoS attributes for associated web services is provided by web service rating agencies, where higher QI value indicates better web service [8]. The rating agencies provide different rating system for each QoS attribute. Therefore, different rating representations are normalized for processing and formulized as following.

$$\text{Quality Index } QI = \left(\frac{\sum_{i=1}^{n} w_k}{n}\right) * W_k \qquad (1)$$

Where, W_i is the index value of an attribute given by the i^{th} agency for a web service, *n* is the number of agencies, and W_k is normalized QI value between 0 and 1. QI value is normalized to keep the range within 0 to 1, so that any higher value of W_i cannot bias the result.

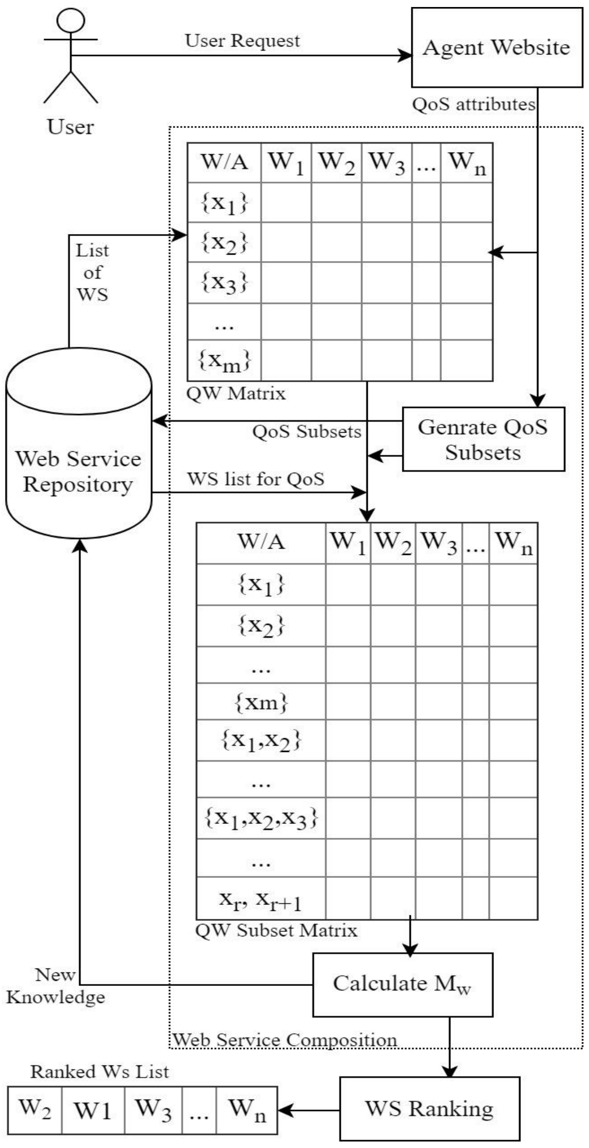

Fig.2. Architectural Overview of SMSR Framework

The standard QI value for each attributes with associated services is shown in Table 2. Where, $QI_{ij} = 0$ and $QI_{ij} < 0$ indicates that i^{th} QoS attribute is not supported by j^{th} web service.

Table 2. QW Matrixes with Standard Index

w/a	W_1	W_2	W_3	…	W_n
x_1	QI_{11}	QI_{12}	QI_{13}	…	QI_{1n}
x_2	QI_{21}	QI_{12}	QI_{23}	…	QI_{2n}
x_3	QI_{31}	QI_{32}	QI_{33}	…	QI_{3n}
…	…	…	…	…	…
x_m	QI_{m1}	QI_{m2}	QI_{m3}	…	QI_{mn}

Algorithm 1: QW Matrix Generation
Input: QoS attributes
Output: QW matrix

```
1:   Begin
2:   $L_{at} \leftarrow$ read all QoS attributes
3:   $L_{ws} \leftarrow$ read all web services in repository
4:   $QW_m \leftarrow [][]$
5:   for all attributes $a_i \in L_{at}$ do
6:       for all web services $w_j \in L_{ws}$ do
7:           if $a_i$ is satisfied by $w_j$ then
8:               $QW_m[a_i][w_j] = 1$
9:           else
10:              $QW_m[a_i][w_j] = 0$
11:          end if
12:      end for
13:  end for
14:  End
```

The symbols L_{at}, L_{ws} and QW_m in algorithm 1, are used as list of QoS attributes, list of available web services and QW matrix respectively. Web services with associated attributes are checked and mapped in QW matrix in line 5 to 13. Where $QW_m[a_i][w_j]$ is the QW matrix index of j^{th} web service with its associated i^{th} attribute.

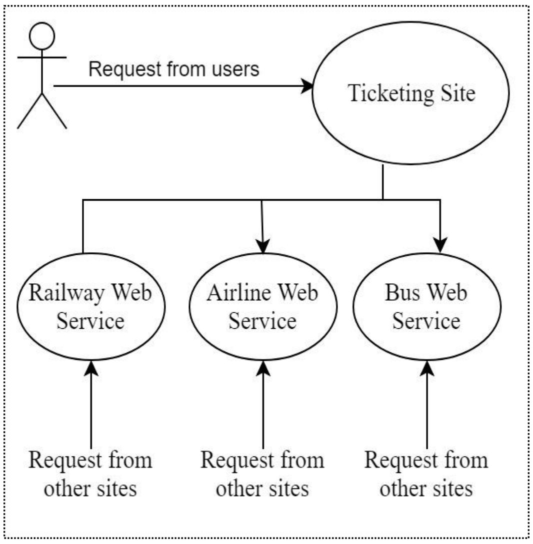

Fig.3. Web Service Request Example

B. Step 2: Generating Subset from QoS Attributes

QoS is a combination of several attributes associated by web services such as availability, security, response time and throughput [12]. The QoS attributes are used by agent programs that send requests to web service providers in order to meet the users' requirements. An example of request program and service provider is shown in "Fig.3", where agent program (e.g. ticketing site) generates required QoS attribute list according to user requirements. These generated QoS attributes are accepted by the SMSR framework which then processed to provide desired web service. At the end of this step, subsets of QoS attributes are generated for creating QW matrix subset. If an agent program submits request with a number of QoS attributes for example $\{x_1, x_2, x_3, \ldots, x_r\}$, the subset of these attributes will be $\{x_1\}$, $\{x_2\}$, $\{x_3\}$, …,$\{x_n\}$,..,$\{x_1, x_2\}$, $\{x_1, x_3\}$, …,$\{x_1, x_2, x_3\}$, …,$\{x_1, x_2, x_3, \ldots, x_r\}$. Where r is the number of QoS attributes. Therefore, total number of subset will be two power number QoS attributes minus one which is defined as follow.

$$\Pr_{subset} = 2^r - 1 \tag{2}$$

C. Step 3: Creating Subset of QW Matrix

Subset of QW matrix represents web services with their associated QoS attributes where subset attributes are entered as rows and web services are entered as columns. The index value (QI_{ij}) in subset QW matrix is given using matching result between i^{th} QoS subset and j^{th} web service. Where, zero value of QI_{ij} indicates no matching found between attributes and web services.

If W_1, W_2, W_3, …, W_n web services provides $\{x_1, x_2, x_3, \ldots, x_m\}$ QoS attributes, generated subset QW matrix and standard QI value will be as shown in Table 3 and functional steps are discussed in algorithm 2.

Table 3. Subset QW Matrix

w/a	W_1	W_2	W_3	…	W_n
$\{x_1\}$	QI_{11}	QI_{12}	QI_{13}	…	QI_{1n}
$\{x_2\}$	QI_{21}	QI_{22}	QI_{23}	…	QI_{2n}
…	…	…	…	…	…
$\{x_m\}$	QI_{m1}	QI_{m2}	QI_{m3}	…	QI_{mn}
$\{x_1, x_2\}$	$QI_{\{12\}1}$	$QI_{\{12\}2}$	$QI_{\{12\}3}$	…	$QI_{\{12\}n}$
…	…	…	…	…	…
$\{x_1, x_2, x_3\}$	$QI_{\{123\}1}$	$QI_{\{123\}2}$	$QI_{\{123\}3}$	…	$QI_{\{123\}n}$
…	…				
$\{ x_r, x_{r+1} \}$	$QI_{\{rr+1\}1}$	$QI_{\{rr+1\}2}$	$QI_{\{rr+1\}3}$	…	$QI_{\{rr+1\}n}$

In this subset of QW matrix, QI value of single attributes (e.g. $\{x_1\}$, $\{x_2\}$, …, $\{x_m\}$) are determined from QW matrix as represented in Table 1. On the other hand, composite QoS attributes (e.g. $\{x_1, x_2, x_3\}$) QI value are calculated from previous subset value in QW matrix. For example, QI value of $QI_{\{x1, x2, x3\}}$ is calculated using previous subsets value as below.

$$QI_{\{x1,x2\}} = QI_{\{x1\}} + QI_{\{x2\}}$$

$$QI_{\{x2,x3\}} = QI_{\{x2\}} + QI_{\{x3\}}$$

$$QI_{\{x1,x2,x3\}} = QI_{\{x1,x2\}} + QI_{\{x2,x3\}} + QI_{\{x3,x1\}}$$

Therefore, value of $QI\{x_1, x_2, x_3\}$ is the summation of $QI\{x_1, x_2\}$, $QI\{x_2, x_3\}$ and $QI\{x_3, x_1\}$ subsets. As the number of subset depends on number of attributes, the QI value calculation can be defined as follow.

$$QI of \{x_1, x_2, x_3, ..., x_r\} = \sum QI(^rC_1) + \sum QI(^rC_2 + ... + \sum QI^rC_r$$

Where,

$QI(^rC_1)$ is $QI(x_1) + QI(x_2) + \ldots + QI(x_r)$ and
$QI(^rC_2)$ is $QI(x_1, x_2) + QI\ (x_2, x_3) + \ldots + QI(x_{r-1},x_r)$

Therefore,

$$QI\{^rC_n\} = \sum QI(^rC_{n-1}) \quad (3)$$

Where, value of n is one to r.

For example, subset calculation will be as below for three attributes.

Subsets of $\{x_1, x_2, x_3\}$ = $\sum 3C_1 + \sum 3C_2 + \sum 3C_3$
Where, $\sum 3C_1 = \{x_1\} + \{x_2\} + \{x_3\}$,
$\sum 3C_2 = \{x_1, x_2\} + \{x_2, x_3\} + \{x_3, x_1\}$, and
$\sum 3C_3 = \{x_1, x_2, x_3\}$

And QI value will be calculated as follow.

$QI\{x_1, x_2, x_3\} = \sum QI(3C_2)$
$= QI(x_1, x_2) + QI(x_2, x_3) + QI(x_3, x_1)$ [According to equation (3)].

Subset of QW matrix value is used to find matching weight of web services, which indicates how number of QoS attributes are satisfied by a web service. The matching weight is the highest *QI* value of a particular web service, which is the summation of highest number of attributes subset QI value.

Algorithm 2: Subset QW Matrix Generation
Input: QoS attributes, QW matrix, WS repository
Output: Subset QW Matrix

```
1:  Begin
2:  $A_{ttr} \leftarrow$ read all QoS attributes
3:  $L_{ws} \leftarrow$ read all web services in repositroy
4:  $QW_m \leftarrow$ QW matrix
5:  $Atr_{sub} \leftarrow \{\}$
6:  $SQW_m \leftarrow \{\}$
7:  for all attributes $a_i \in A_{ttr}$ do
8:      for all services $ws_j \in L_{ws}$
9:          if $a_i \in QW_m$
10:             $SQW_m[a_i][ws_j] = QW_m[a_i][ws_j]$
11:         else
12:             $Atr_{sub} \leftarrow$ subset of $a_i$
13:             for subsets $Asub_j \in Atr_{sub}$
14:                 temp += $QW_m[Asub_i][ws_i]$
15:             end for
16:             $SQW_m[a_i][ws_j] = temp$
17:         end if
18:     end for
19: end for
20: End
```

In algorithm 2, QoS attributes and web services are stored in A_{ttr} and L_{ws} (line: 2-3). QW_m is initialized with previously generated QW matrix (line: 3). Subset QW matrix is generated between line 7 to 19. If attributes belong to QW matrix, the subset is initialized with it (line: 9-10). On the other hand, subsets of attributes are generated and values of all subsets are added to initialize subset QW matrix (line: 12-16).

D. Step 4: Selecting and Ranking Procedure

Selecting and ranking of web services for composition is performed using obtained matching weight of each web service in previous section. In this step, web services with matching weight greater than a threshold value are selected and considered for ranking. Web service with higher matching weight (M_w) gets higher ranked value and selection priority. Thus, a number of web services are listed according to matching weight and provides to the requested program (agent site). If the result is being accepted, the new QoS attributes with its related web services are updated in WS repository as a new knowledge. The web service will be randomly selected if more multiple services have the same matching weight.

The proposed subset matching based web service composition approach is implemented to select and rank web service for satisfying the user requirements. This strategy generates subset from QoS attributes to create QW subset matrix and finding matching weight of every web services. The web services with higher matching weight is selected and ranked top in the list. In addition, if a set of QoS attributes satisfy agent program, the matching weight of those attributes are updated in WS repository. The overall architectural view of proposed system is presented in Fig. 2 and functional steps are demonstrated in algorithm 1 and 2 accordingly.

IV. Evaluation

The environmental setup, experimental result analysis and investigated result comparison has present in this section to evaluate the proposed SMSR strategy. The effectiveness of proposed subset matching based web service ranking has also been evaluated and reported herewith. Web service composition time and selection accuracy are the two key factors for web service ranking method which are considered to show effectiveness of the proposed system.

A. Environmental Setup

This research work evaluation has performed on personal computers having 2.5 GHz core i5 CPU and

8GB memory running the windows 7 operating system. The simulation environment has developed with Visual Studio 2008, ASP.NET 3.5, XML Web Service, .NET Framework 3.5, MySQL Server 5.0, MySQL Connector Net 5.1.6, Windows Server 2003, IIS 6 (Internet Information Services).

B. Simulation

The simulation environment was developed with several existing web services which are kept on local and remote network. Those services are considered to get real time experience. The experiments were repeatedly conducted 40 times and the average results are reported for consideration. To show simulated result of the proposed SMSR method, several QoS attributes [8] and locally developed web services were used.

C. Result Analysis

For various number of web services composition time is measured to show its complexity in respect to required time. In addition, web service composition accuracy has shown for various numbers of web services for discussing comparative results.

Time Complexity: Web service composition time against the number of services indicates the effectiveness of web service selecting and ranking strategy. The time complexity of proposed system is compared with existing SVD based system [8]. SVD technique is well established for web service decomposition. The complexity increases with the incremental number of web services and associated QoS attributes. Computational complexity of SVD to decompose a two dimensional matrix is calculated as follow.

$$f(N,K) = O(N^2k + Nk^2 + k^3) \tag{4}$$

Where N is number of columns and k is number of rows. When N is always greater than K the complexity will be as below.

$$f(N,k) = O(Nk^2) \tag{5}$$

In contrary, proposed subset matching algorithm requires total comparison as follow.

$$\begin{aligned} T_{com} &= m(2^r - 1)(2^n - 1) \\ &= m(2^{r+n} - 2^r - 2^n + 1) \end{aligned} \tag{6}$$

Where r is total number requested attributes, n is maximum number of supported QoS attributes of a web server and m is the total number of web services.

Sometime maximum r can be equal to n. Therefore, when r is equal to n, the number of comparisons

$$\begin{aligned} &= m(2^{n+n} - 2^n - 2^n + 1) \\ &= (m2^{2n} - 2^{n+1} + 1) \end{aligned} \tag{7}$$

Here, $2^{2n} > 2^{n+1}$, so complexity will be $O(2^{2n})$, which is better than $O(Nk^2)$, therefore, proposed method works faster than existing SVD, which is shown in the following simulation result Table 4.

Table 4. Response Time

WS	Response Time	
	SVD	SMSR
10	0.62	0.48
20	0.702	0.503
30	0.748	0.56
40	0.802	0.58
50	0.85	0.59

Data in Table 4 demonstrates the response time of SVD and SMSR techniques for different number of web services and plotted in “Fig.4”, where required service response time is always smaller for SMSR system compared to SVD.

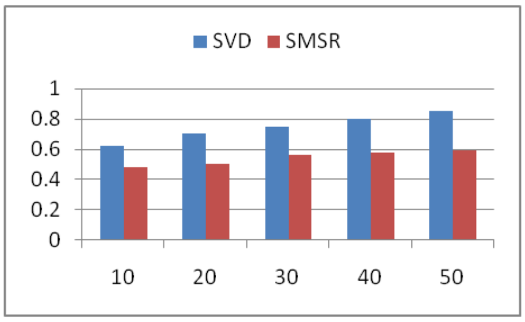

Fig.4. Response Time for Web Services

In this table, response time of SVD for 10, 20, 30, 40 and 50 web services are 0.62, .0702, 0.748, 0.802 and 0.85 second respectively. Where response time of SMSR for 10, 20, 30, 40 and 50 web services are 0.48, 0.503, 0.56, 0.58 and 0.59 second respectively which is always less than existing system.

The web service response time varies when number of input attributes is changed. The response time for various number QoS attributes are shown in Table 5, where attributes and response time are represented horizontal and vertical cells respectively.

Table 5. Response Time for No. of Attributes

No. of Attributes	Response Time	
	SVD	SMSR
1	0.45	0.3
2	0.5	0.302
3	0.54	0.3285
4	0.5406	0.3438
5	0.5406	0.3438
6	0.5406	0.3438
7	0.5406	0.3438

The response time of proposed method and SVD techniques for different number of attributes is shown in “Fig.5”. In the figure response time of proposed method is always lower compared to existing system. In SVD, the response time ranges from 0.45 to 0.54 second for attribute range of 1 to 7. However, response time of SMSR ranges from 0.3 to 0.34 second for attributes range of 1 to 7 which indicates lower value than SVD.

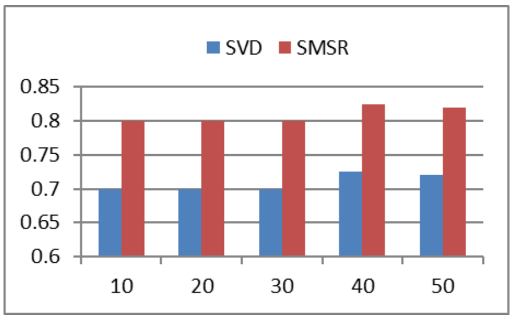

Fig.5. Response Time for Attributes

Accuracy Comparison: Accuracy of the proposed framework is calculated as the ratio of the relevant and total number of selected web services. The effectiveness of the selection process between the proposed SMSR and existing SVD system is measured in terms of accuracy, precision and recall. Accuracy is calculated by dividing the number of correctly predicted instances (in this case web services) by the total number of instances as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Where TP = true positives (the web services those are correctly selected), TN = true negative (the web services those are correctly discarded), FP = false positive (the web services those are selected but actually not correct to request), FN = false negative (the web services those are not considered for selection but actually relevant to request) and TP + TN + FP + FN = total number of web services. Therefore, accuracy of correctly selected web services is calculated as:

$$Accuracy = \frac{\text{Selected WS} + \text{Discarded WS}}{\text{Total WS}} \tag{9}$$

Total 50 web services have been taken as experiment to measure the accuracy of proposed and existing system. For a request, list of recommended web services of the proposed and existing system has collected. Then, relevant and irrelevant services are profiled and compared with the list of recommended services for both SMSR and SVD methods. From the comparison results true positive, false positive, true negative, false negative is counted to find out the comparative accuracy result.

Table 6.Web Services Selection

WS	Selected		Discarded	
	SVD	SMSR	SVD	SMSR
10	7	8	3	2
20	15	17	5	3
30	25	25	5	5
40	32	34	8	6
50	38	42	12	8

The experimental results in Table 6 shows that for 10, 20, 30, 40, 50 web services SVD method selects 7, 15, 25, 32, and 38 web services respectively, where proposed SMSR method selects 8, 17, 25, 34, and 42 web services respectively which is plotted in "Fig.6".

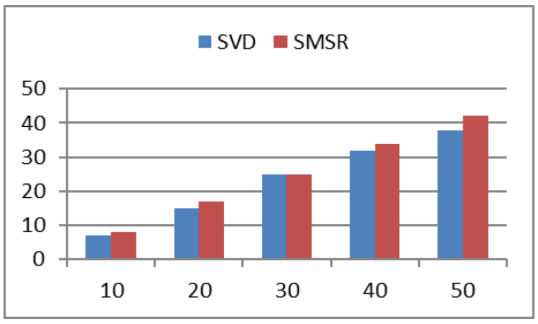

Fig.6. Web Service Selection

Out of selected 7 web services for SVD method 5 are relevant and out of discarded 3 services 2 are relevant, so for this case: true positive and true negative value is 5 and 2 respectively. On the other hand, false positive is 2 and false negative is 1. Therefore, according to equation 9 accuracy of existing system is (5+2)/10 or 0.7. Accuracy of SVD method for 20, 30, 40 and 50 web services has measured accordingly and shown in Table 7.

Table 7. Web Services Selection Accuracy

WS	SVD		SMSR		Accuracy	
	TP	TN	TP	TN	SVD	SMSR
10	5	2	6	2	0.7	0.8
20	11	3	14	2	0.7	0.8
30	18	3	21	3	0.7	0.8
40	24	5	29	4	0.725	0.825
50	30	6	36	5	0.72	0.82

In Table 7, accuracy values of SVD technique for 10, 20, 30, 40 and 50 services are 0.7, 0.7, 0.7, 0.725 and 0.72 respectively. Where, accuracy values of proposed SMSR method for 10, 20, 30, 40 and 50 services are 0.8, 0.8, 0.8, 0.825, and 0.82 respectively. Using both the existing and proposed system's accuracy results are plotted in "Fig.7", which clearly indicate that proposed SMSR based method has better accuracy level than existing SVD method.

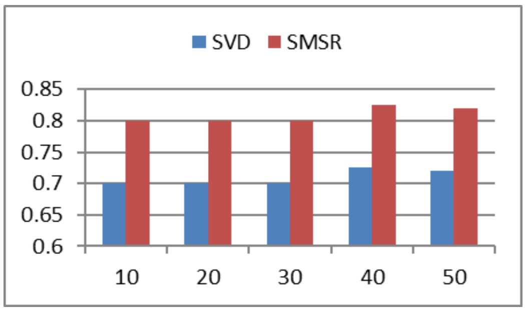

Fig.7. WS Selection Accuracy

From the above experimental results and comparative study, it demonstrated that proposed subset matching based web service selection exhibits better performance than the existing system considering the features of composition time, service selection accuracy, rejection accuracy and scalability in terms of number of QoS attributes and web services.

E. Discussion

Compared to the SVD based approach, proposed SMRS method reduces time complexity. Where SVD based decomposition for a two dimensional QW matrix with *N* columns and *k* rows, the complexity is $O(Nk^2)$, and in worst case if *k* become equal to *N*, the complexity is $O(N^3)$. On the other hand, proposed SMSR method generated a subset QW matrix with $2^r\text{-}1$ columns and *N* number of rows, where *r* is the number of required QoS parameters. In the worst case, the maximum number of *r* can be equal to *N* and complexity is $O(2^{2N})$. Therefore, comparing these two factors, it is clear that the proposed method is better than the existing method. There is no need of data dimensionality reduction in the proposed method and that is why it shows the better accuracy than SVD based approach.

V. CONCLUSION

This paper proposed a subset matching based web service selection and ranking framework using QoS attributes. The framework improves the performance of web service recommendation process. Web service repository and QoS attributes are used to select relevant services and stored in QW matrix. When an agent site send request with QoS attributes, the framework creates subset of QW matrix using requested attributes. Ranking process is performed using individual weighted web services from QW matrix subset.

The proposed method is evaluated with two key factors named time complexity and accuracy. The composition time has been measured and found that SVD method for different number of web services response time weighted score was 0.78 seconds. Where for same number of web services SMSR technique's response time weighted score was 0.56 seconds, which is 39% better than existing SVD system. As well as, selection and rejection accuracy for various number of web services are measured and found that SMSR method performs 12.30% better than existing system. The investigated results reveal that the uses of subset matching approach improves the efficiency of web service ranking in terms of response time and accuracy. The main contributions of this paper are: (i) introducing a novel subset matching approach for web service ranking and (ii) improved accuracy and scalability for web service composition. The future direction of this research will be applied several stochastic algorithms to select the service when unknown input attributes arises.

REFERENCES

[1] Amira Abdelatey, Mohamed Elkawkagy, Ashraf Elsisi, Arabi Keshk,"Extend Web Service Security Negotiation Framework in Privacy", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.8, pp.30-39, 2017. DOI: 10.5815/ijitcs.2017.08.04

[2] Neiat AG, Bouguettaya A, Sellis T. Spatio-temporal composition of crowdsourced services. InInternational Conference on Service-Oriented Computing 2015 Nov 16 (pp. 373-382). Springer, Berlin, Heidelberg.

[3] Maamar Z, Benslimane D, Mostéfaoui GK, Subramanian S, Mahmoud QH. Toward behavioral web services using policies. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans. 2008 Nov; 38(6):1312-24.

[4] Aslam MA, Shen J, Auer S, Herrmann M. An integration life cycle for semantic web services composition.Proceedings of the 11th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2007) (pp. 490-495). Melbourne: Swinburne Press.

[5] Randa. Hammami, Yossra. Hadj Kacem, Senda. Souissi, Hatem. Bellaaj, Ahmed. Hadj Kacem ,"Weighted Priority Queuing: A New Scheduling Strategy for Web Services", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.2, pp.11-17, 2017. DOI: 10.5815/ijitcs.2017.02.02

[6] Sirin E, Parsia B, Hendler J. Filtering and selecting semantic web services with interactive composition techniques. IEEE Intelligent Systems. 2004 Jul 1(4):42-9.

[7] Maamar2 Z, Mostefaoui SK, Yahyaoui H. Toward an agent-based and context-oriented approach for web services composition. IEEE transactions on knowledge and data engineering. 2005 May;17(5):686-97.

[8] Chan H, Chieu T, Kwok T. Autonomic ranking and selection of web services by using single value decomposition technique. InWeb Services, 2008. ICWS'08. IEEE International Conference on 2008 Sep 23 (pp. 661-666). IEEE.

[9] Kwon J, Kim H, Lee D, Lee S. Redundant-free web services composition based on a two-phase algorithm. InWeb Services, 2008. ICWS'08. IEEE International Conference on 2008 Sep 23 (pp. 361-368). IEEE.

[10] Lamparter S, Ankolekar A, Studer R, Grimm S. Preference-based selection of highly configurable web services. InProceedings of the 16th international conference on World Wide Web 2007 May 8 (pp. 1013-1022). ACM.

[11] Hachemi, Yamina & Benslimane, Sidi Mohamed. (2015). Preference-Based Web Service Composition: Case-Based Planning Approach. International Journal of Information Technology and Computer Science. 7. 74-82. 10.5815/ijitcs.2015.06.10.

[12] Menascé DA. QoS issues in web services. IEEE internet computing. 2002 Nov;6(6):72-5.

Authors' Profiles

Md. Abdur Rahman received his BSc in Information Technology from Visva Bharati University, India in 2004. He has completed his Post Graduate Diploma and Master in Information Technology from University of Dhaka, Bangladesh, in 2008 and 2009 respectively. He is a Senior Computer Scientist in the Centre for Advanced Research in Sciences at the University of Dhaka. His major research interest includes natural language processing, machine learning, deep learning, big data analytics and software engineering. He has published a number of research papers in various international journals and conferences.

Md. Belal Hossain was born in Kushtia, Bangladesh, in 1982. He received the B.Sc. degree in Computer Science & Engineering from the Khulna University, Bangladesh, in 2008 and the Master in Information Technology from the University of Dhaka, Bangladesh, in 2010. He is now working at The Central Bank of Bangladesh as a software engineer. His major research interest includes cyber security, smart & secured application development, search engine optimization, data recovery algorithms. He has published a number of research papers in various international conferences.

Md. Sharifur Rahman received his Post Graduate Diploma and Master in Information Technology from University of Dhaka, Bangladesh, in 2008 and 2009 respectively. He is Senior IT Consultant Government Republic of Bangladesh Finance Division, Ministry of Finance. His major work area includes Government Financial Management, Pension & Fund Management, Budget Management analytics, Service Oriented Architecture, Web Service Management and software engineering.

Saeed Siddik have been working on Software Testing and Software Analysis research where he experimented how software are developed and tested efficiently. He has completed his M.Sc. in Software Engineering, including the highest marked thesis dissertation on Software Test Case Prioritization from IIT University of Dhaka. The research outcomes of that thesis were published at several Journal and Conferences. He was the first research student of IITDU Optimization Research group, where he was working on software design migration to enhance modularity and manageability. He is a member of IEEE (ID:94159542).

How to cite this paper: Abdur Rahman, Belal Hossain, Sharifur Rahman, Saeed Siddik, "Subset Matching based Selection and Ranking (SMSR) of Web Services", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.4, pp.44-53, 2019. DOI: 10.5815/ijitcs.2019.04.05