MAIL SERVER FOR ORGANIZATION

1.KASTHURI.S 2.BHAVANI.D 3.SURIYA.S

B.E-FinalYear/CSE,Gnanamani College of Technology,Namakkal-637018,Tamilnadu.

Abstract— Selection of an appropriate web service fulfilling the requirements of the end user is a challenging task. Most of the existing systems use Quality of Service (QoS) as predominant parameter for web service selection, without any preprocessing or filtering. These systems consider all of the candidate web services during selection process and require unnecessary processing of those web services which are far below the expectations of the end user. In this work, an approach for web service selection based on QoS parameters is proposed. The proposed method starts with prefiltering of candidate web services using classification technique. An improved PROMETHEE method, we call it as PROMETHEE Plus, is applied to most eligible web services and Maximizing Deviation Method based hybrid weight evaluation mechanism is adopted. Top-k web services matching closely with the QoS requirements of the end user are selected. Experiments on the dataset of real world web services are conducted. Experimental results show that our approach performs better in terms of end user satisfaction and efficiency with reference to the existing similar approaches.

Index Terms— Classification, Maximizing deviation method, PROMETHEE, QoS, Web Service, Web Service Selection.

1INTRODUCTION:

SELECTION of most promising Web Service (WS) satisfying needs of the end user is a core and challenging task in Service Oriented Architecture. With the increasing use of technologies like cloud computing, Internet of Things, etc., a lot of WS providers are offering their business functionality as WS. This has resulted in an ex-ponential increase in the number of functionally similar WSs with diverse QoS. Hence, the selection of desired WS from multiple competing services offering same function-ality is a difficult task. The task of selection is usually per-formed in two steps. First, from the WS pool, services offering identical functionality are fetched. In this regard, existing solutions such as AI planning based methods [1] or semantic based approaches [2],[3],[4] are efficient. Next, from the list of functionally similar WSs, a WS is to be selected. Usually, QoS parameters are used to differen-tiate among functionally equivalent WSs. The QoS parameters such as availability, response time, reliability, throughput, documentation, etc., are among the few popular QoS parameters [5]. The QoS parameters collectively characterize the quality a service offers. The web services with same functionality may exhibit different quality. To understand this idea, we have identified few services from weather and hotel domain from QWS dataset [5]. Weather service determines the current weather condition and hotel service offers the facility such as, hotel booking, the current booking status of hotel, etc. Based on the throughput, response time, and availability QoS parame-ters, quality distribution of few services of hotel and weather domain is drawn and shown in Fig. 1. It can be analyzed from the figure that few of the WSs are best in all three QoS parameter values, while few other services are having low response time but have low availability and low throughput. Few of the WSs have moderate qual-ity. From the figure, it is clear that the web services offer similar functionality, but differ in their QoS values. There-fore, QoS parameters can be regarded as a useful criterion by the web service selection (WSS) system to discriminate two WSs. Further, the WSS system also measures, how well the QoS requirements of the end user are satisfied by the candidate WSs. This measure is used to rank the can-didate services.

The process of creating the relative ranking of WSs based on QoS parameters can be reduced to the problem of decision making. The Multi Criteria Decision Making (MCDM) methods are suitable to solve such problems [6]. The Preference Ranking and Organization METHod for Enrichment Evaluation (PROMETHEE-II) is a popular MCDM method used in past for WSS [6],[7]. There are other available approaches for efficient WSS such as using the Probabilistic QoS based WSS [1], Skyline technique [8], [9], [10], Utility function based selection [11], Mix In-teger Programming (MIP) [12],[13], Case Based Reasoning (CBR) [14], Collaborative filtering [15], and service bun-dling [17],[18],[25].

The QoS parameters are conflicting and difficult to compare. These conflicts among QoS parameters lead to interdependency between them, which cannot be ignored during service selection. Existing solutions [1],[9],[12],[15] lack in terms of considering the conflicting nature of QoS parameters, which should be included in the selection step to produce accurate results. Moreover, the perfor-mance in terms of time of the existing techniques includ-ing available MCDM techniques [6],[7],[16] deteriorates as more and more functionally equivalent WSs are made available. Because, more efforts are required for selection of WS and hence overall performance of WSS system de-grades. To improve the performance of WSS system, the selection is done in two steps [8],[10]. Firstly, WSs having higher end QoS is obtained using a skyline technique. The skylined services are prioritized in second step. Nonethe-less, the skyline technique has its own issues [9]. The sky-line approach select same list of services irrespective of the end user requirements. It causes the same set of ser-vices to be presented before different end users, inde-pendent of their QoS requirements. Therefore, in addition of ignoring the end users requirements, it also causes problem of imbalance of load on a specific set of services [26]. The end users requirements during evaluation are also ignored in existing MCDM based WSS models [7], [8]. Further, the end user specifies importance of QoS by using weight values. The existing solutions such as [1],[10],[11],[13] allows end users to provide numerical values to represent the QoS importance. The numerical value of weights are difficult to comprehend for a naïve user. Also, the use of techniques such as AHP [8], ANP [20], etc., for QoS weight evaluation, require input from domain experts. It makes the WSS system domain de-pendent and inflexible.

Based on the gaps discussed above, the research prob-lem can be summarized as follow:

Research problem: Let, *S = {S1,S2,.........,Sn}*: Set of *'n'* discrete web services offering same functionality; *QV = { , ,..........., }*: represents a QoS vector with set of *'m'* conflicting QoS parameters for decision making; *WSQoS*: is a matrix representing the set of all QoS parame-ters for S; *UserQoS*: is a Web Service representing the QoS requirements of the end user. The challenge is to select most optimal WS *Sr* from the set *S* such that with three considerations. Firstly, the size of set *S* is in-creasing as more and more WS providers are offering WSs with the same functionality. Secondly, the QoS pa-rameters are conflicting in nature. Third, the system should support a mechanism to evaluate weight of QoS parameters using mathematical model. Moreover, the weight evaluation mechanism should allow the end user to specify partial preference of QoS. Therefore, an effi-cient technique is needed to solve the problem of WSS with due concern to above three criteria. Selection of top-k services require identification of *K* such services which closely meets .

Research solution: The conflicting nature of QoS parameter can be considered during service selection step. A PROMETHEE based solution can be helpful in providing consideration to conflicting nature of QoS parameters [6]. To improve the end user satisfaction of QoS, PROMETHEE method for WSS can be modified in three ways. Firstly, the end user request of QoS can be included during the evaluation phase of PROMETHEE. Second, those WSs which are close to can be identified. Third, the choice of evaluation function can be made as per the type of QoS parameter. To improve the perfor-mance of service selection system, functionally similar WSs can be preprocessed. Two conditions must be satis-fied by the preprocessing based WSS system. Firstly, the end user requirements must be considered to avoid non-compliance of the user requirements. Secondly, the pre-processing system should be independent of selection system and should filter out only non-eligible services. Eligible WSs should remain in the system for further pro-cessing by selection module. PROMETHEE method offers flexibility to specify weight of QoS parameters externally. MDM based mathematical model is useful to evaluate weight values [22]. The weight preference specified by the end user can be combined with the QoS weight obtained from mathematical model.

The motivational example to better explain the motiva-tion underlying our proposed design is discussed in sec-tion 3.1. Following are contributions of this paper:

1. For ranking of web services with due considera-tion to conflicting QoS parameters, an improved PROMETHEE approach called as PROMETHEE Plus is proposed. This improves the end user satis-faction and also the quality of WS(s) to be selected.
2. A hybrid QoS weight evaluation scheme based on Maximizing Deviation Method (MDM) is suggest-ed to be used with PROMETHEE Plus method. It causes the WSS system to be domain independent.
3. Classification based prefiltering technique is de-veloped to improve the performance of WSS algo-rithm.
4. Statistical analysis is done to observe the system behavior and compared with existing approaches of WSS.

Performance of the proposed system is measured us-ing Euclidean distance, query hardness, satisfaction score and time for web service selection. The experimental analysis performed using these parameters shows that the proposed CSS approach has outperformed over existing similar approaches. It is observed that the use of PROMETHEE Plus approach provides improvement in the end user satisfaction by including the end user re-quest during service evaluation and handling the conflict-ing QoS parameters. Further, the use of classification based prefilter layer improves the performance of WSS system and reduces overall selection time. The use of MDM based weight evaluation approach makes WSS sys-tem more robust and domain independent.

The rest of the paper is organized as follows. Existing works related to WSS are discussed in Section 2. In Sec-tion 3, proposed approach for WSS and system architec-ture along with the algorithm for WSS is presented. The experimental analysis and results are discussed in set.

2 RELATED WORKS

Few existing solutions related to PROMETHEE based WSS, QoS parameters weight evaluation, and WS Classi-fication are discussed in this section.

QoS is being used as an important parameter to distin-guish among functionally similar WSs. It is increasingly difficult and time consuming task to obtain best WS due to conflicting QoS parameters. An existing QoS based WSS approach treats QoS parameters as discrete random variables with probability mass functions [1]. The service selection is performed by computing the utility function based on the probability of satisfying QoS constraints. A similar approach for WSS using utility function for sort-ing and selection of web services is available [11]. In this approach, the distance between the service candidate and the upper bound is obtained. Based on distance values, candidate WSs are sorted. Nevertheless, the approach does not consider the conflicting nature of QoS parame-ters during evaluation of the utility function, which we believe are very crucial for producing an accurate ranking of services. Web Service Relevancy Function (WsRF) can also be used for web service ranking and selection [5], [20]. Higher value of relevancy function indicates better service. Similar to our work, variety of solutions for QoS based WSS problem are available such as – WSS based on constraint rules using MIP [12], similarity rules in CBR [14], predictions based on collaborative filtering [15], etc. However, these solutions do not consider the conflicting nature and interdependency among QoS parameters.

An effective method for comparison of alternatives based on the conflicting QoS attributes is PROMETHEE-II [6]. It performs total ordering of alternatives. The PROMETHEE method for WSS is introduced by [21]. The QoS parameters are mapped to the actions and service providers are the alternatives. To define the criteria, ap-propriate choice of preference function is to be made. For qualitative criteria (performance, availability), usu-al/Gaussian type preference functions are suitable. Simi-larly, for quantitative criteria (cost, price, and power), V-shape/level/linear type preference functions are suitable [6]. The preference index of services needs to be evaluated by pair wise comparison of alternatives. The net outrank-ing flow for each candidate WS is calculated, which re-sults in a ranking of WSs [16]. The PROMETHEE method does not include any predefined weight calculation scheme. The weight calculation is kept flexible. The modi-fied Simos procedure for calculation of QoS weights is one such method to specify the QoS weight externally [7].

The weight of the QoS parameter is one of the factors which guide the performance as well as output of WSS method, hence it cannot be ignored. In the literature, it is assumed that the end user has a clear idea about QoS preferences and is able to assign a scalar value to repre-sent the QoS preferences [5], [9], [11], [12], [29], [31]. On the contrary, this is rather a big challenge for the end user to properly judge the QoS weight values and for WSS module to provide the best selection by incorporating the user judgment. In the available works, methods such as AHP [7], [8] and ANP [16] are used to represent the pref-erence of the end user. However, the preference is deter-mined by taking input from the domain experts. A more useful way to determine the linguistic weights of QoS parameters is based on a resolution process in which a group of participants' preferences is considered [23]. A more natural way to give preference to the user desired QoS parameters is to include them in the calculation. The skyline based approach for WSS does not require QoS preference from the end user explicitly [24]. The proposed approach in this paper uses MDM method for evaluation of weights of QoS parameters.

WSs can be grouped together by performing classifica-tion task. Classification of WSs can be achieved by using information in WSDL [34], [35], OWL-S [33], QoS [36], [37] and others. The available works on classification using WSDL and OWL-S performs classification at the time of service discovery. As it is performed at the time of service discovery, so, they are increasing the efficiency of discov-ery process, generating a set of functionally similar WSs. To further reduce the set of functionally similar WSs, our work is performing classification of WSs based on associ-ated QoS information. Thus, classification is applied in prefiltering step to reduce domain of search. A similar approach based on bundling framework to reduce search domain is presented in [17],[18],[25]. The presented bun-dling framework uses complementarity indexes and user context to create service bundles. The customer and ser-vice providers interact with each other to identify the cus-tomers' need. Upon clearly identifying the needs of end user, service bundles are generated dynamically. The ser-vice clusters are created based on the services offering similar functional consequences (FCs) [17],[18],[25]. This method achieves increased user satisfaction, social wel-fare and customer surplus. This method can be useful for efficient selection of WSs for composite service. However, our proposed approach concentrates on selection of ser-vice for atomic task.

In this work, we have proposed WSS approach which considers conflicting nature of QoS parameters during services selection. The weight evaluation of QoS parame-ters is made domain independent by the use of MDM method. Further, the classification based prefiltering is proposed to be included before selection step. The prefiltering is added to reduce the number of candidate WSs to be processed in selection step. Appendix-A2 summarizes the comparison of our proposed approach with existing similar works.

3 PROPOSED APPROACH FOR WEB SERVICE SELECTION

In this case, QoS requirements of three users' are same. The response to the query contains WSs satisfying the user query also shown in Table 2. The list of WSs satisfy-ing user query is determined using Simple Additive Weight (SAW) [19]. Although, QoS requirements for all three users are same, but they differ in weight values of QoS parameters. The list of WSs satisfying the end user request is different for all three users. User-1 and User-2 minutely differ in their QoS weight values, but the order of services returned to them is different (service S2 and S6). For User-3, weights of cost and RT parameters are largely different as c Further, the existing systems for QoS based WSS re-quire minimum two user inputs: (1) QoS preference (2) Weight of QoS parameters. In most of the cases, end user experiences problems in specifying (1) and (2). The end user can form the query to specify the QoS preference by using user centric system [10] or using the QoS browser [30]. However, weights of QoS parameters need to be judged by the end user. The imprecise specification In the present scenario, the WSS system performs the task of selection in two steps. In first step, the relative ranking of WSs is done. In second step, the selection is accom-plished from the ranked list of WSs. As discussed in sec-tion 2, the basic PROMETHEE method is helpful in ob-taining QoS based ranking of WSs. In this work, we have proposed three variations of WSS system. All three varia-tions are based on an improved PROMETHEE method, PROMETHEE Plus, for ranking followed by top-k selec-tion of WSs. The PROMETHEE Plus method is derived from original PROMETHEE with three modifications. First, the basic PROMETHEE algorithm for WSS is im-proved by including the end-user's request of QoS as part of PROMETHEE evaluation. This improvement will ame-liorate the ranking results by segregating the WSs into two groups. One group will include WSs which exactly satisfy the required QoS and/or have QoS close to the re-quired QoS. The other group includes the WSs having QoS below the QoS expectations of the end user. Second modification is done as part of ranking process. The high-er preference is given to the WS with QoS score closely matching with the requested QoS. This will increase the user satisfaction in using the system [5], [20]. Third, the Gaussian and level type preference functions are used during QoS parameters evaluation. PROMETHEE Plus is discussed in more details in section 3.3.2.

In the traditional WSS system, user could specify the preference of QoS parameters using weight values. The user must provide the precise weight values for all QoS parameters. However, in actual practice, two variations are possible. Firstly, no weights of QoS parameters are provided by the end user. Secondly, partial weights are specified by the end user. In second variation, system should be In this section, initially a motivating example is discussed. Subsequently, the proposed system architecture and algo-rithms are elaborated in detail.

3.1 Motivating Example

QoS weights may result into missing the user desired ser-vices.

The significance and effect of weights of QoS parame-ters for WSS can be understood with the help of an exam-ple. Consider that there are 10 candidate WSs, S1 to S10, with given cost and response time (RT in msec) as shown in Table 1. The cost is measured on the scale of 1 to 10 and RT on the scale of 0 to 1. Lower the values of cost and RT better is the service quality. Consider three user requests arrives in the system with query "Search the WS with Cost $\leq$ 6 and RT $\leq$ 0.5 with QoS weights ompared to User-1 and 2. So, the re-sultant list of WSs is very different for User-3. Hence, it shows that the response of WSS system is largely de-pendent on weight-values of QoS specified by the end user. We could analyze from this example that the per-formance and results of WSS method are highly depend-ent on the QoS weight

3.2 Web Service Selection Approach

flexible enough to encorporate the weight val-ues provided by the end user/expert and weights calcu-lated using mathematical model [22]. In the proposed approach, the first variation is realized by using **M**DM method for weight evaluation in presented PROMETHEE Plus based W**SS** (MSS) and is shown in Fig. 2. A hybrid scheme that combines the user specified weight prefer-ence with the output of MDM method can be used to support partial preference of weights from the end user. This variation using **H**ybrid scheme of weight evaluation in PROMETHEE Plus method for W**SS** (HSS) is presented in Fig. 3. Proposed approaches, MSS and HSS, are practi-cal approaches and useful in scenario where user is a na-ïve and is unable to specify the QoS weight values.

Moreover, if candidate WSs are very large in number, the complexity of the system will also be high. Therefore, a mechanism such as prefiltering of services using classi-fication is proposed as a third enhancement es.

3.3 The proposed CSS approach for WSS

Web service can be labelled using the associated QoS information. Each label associated to the web service rep-resents a pre-defined category. This categorization leads to identification of WSs with similar QoS offerings. The process is referred as classification of WSs. The process of classification starts by creating the learning model using a set of labelled WSs. In the later stage, built model is used to categorize unlabelled WSs. The process of classification of web services helps in reducing the search space. The system architecture of proposed CSS approach is shown in Fig. 4. It consists of two layers. The upper layer is a prefilter layer and bottom layer is selection layer.

4 EXPERIMENTAL ANALYSIS

The proposed approach is tested by conducting different experiments to evaluate the performance and effective-ness in web service selection over existing approaches. In our proposed system, we have considered that all the WSs are functionally equivalent. So, the system needs to perform WSS based on QoS requirements only. The pro-posed WSS approach mainly emphasizes selection of ser-vice for atomic task where service equivalence is more important as compared to compatibility. Details of design of experiments and experimental results are discussed in the following section.

4.1 Experimental Setup

CSS algorithm is implemented in Java and Weka API is used to implement vote based classification. Experiments related to evaluation of the proposed WSS method and its variations are conducted on machine with Intel Core i7 CPU @ 3.4 GHtz, 8GB of RAM, Windows 7 platform and Netbeans installed on it.

4.1.1 The dataset and design of user queries

The QWS dataset (henceforth called as Dataset-1) of real world web services is used to conduct experiments [5],[20]. The dataset consist of 364 labelled and 2507 unla-belled web services. The QoS parameters are evaluated using the QWS measures obtained from Web Service Bro-ker Framework. Nine QoS parameters are used - response time (Rtm), availability (Ava), throughput (Thr), successabil-ity (Suc), reliability (Rel), compliance (to WSDL descrip-tion) (Com), best practice (by following WS-I) (Bpr), latency (Lat) and Documentation (Doc). The last two values in Da-taset-1 represent the service name and reference to the WSDL document. The dataset with 364 web services has additional information of WsRF score and class labels. The WsRF is web service relevance function representing relevance of web services based on QoS parameter values. Using WsRF score, each of 364 web services are assigned a class label out of bronze, silver, gold and platinum. The bronze class represents web services with lowest overall QoS value. Whereas, web services of platinum class have highest QoS value.

The system for web service selection bears the respon-sibility of responding to user query as efficiently as possi-ble. The WSS system responds to users differently as per the user query. In order to test the system behavior and response, hundred queries are generated randomly. Que-ry represents QoS concerns of the end user. Each of the hundred query belongs to a hardness level from one (L1 – least hard) to ten (L10 - hardest). The hardness level L1 represents range from 1-10 (in %), L2 has range of hard-ness from 11-20, and so on. One query from each level is selected and presented in the Table 5. The hardness of each query is obtained using “8,“ and “9,“.

4.1.2 Performance parameters

Three performance parameters are used to evaluate pro-posed and existing approaches. The hardness level, satis-faction level, and Euclidean distance parameters are used for evaluation purpose and defined in the following sec-tion. Time taken for selection of top-k WSs is also used to evaluate each approach. We have selected these parame-ters because these parameters are widely used in the ex-isting similar works for the evaluation of WSS approaches [13],[21],[29],[31].

Hardness: The hardness of a QoS constraint is the ratio of the required QoS to the maximum value of the corre-sponding QoS parameter for the available web services. The hardness of a QoS parameter can be defined using “8,“ [13].

Where, is the maximum value of the QoS parame-ter and is the QoS parameter of WS. Equa-tion (9) is used to determine the hardness of a web service based on the hardness of all associated QoS parameters by taking mean of hardness of all QoS constraints.

Satisfaction: The service satisfaction is defined as a meas-ure of how well the QoS of service meets the users’ con-cerned QoS requirements. The service satisfaction can be obtained from satisfaction score of individual QoS pa-rameter of the service using “10,“ and “11,“ [31].

4.1.3 PROMETHEE Plus parameters evaluation

The PROMETHEE Plus method has six predefined functions to evaluate QoS parameters for comparison among the candidate web services. Preference function

determines preference of one web service over others. For QoS parameters which are quantitative type, such as cost, price, documentation, compliance, etc., level/linear type preference function can be used [21]. The parameters which are qualitative in nature, such as reliability, availab-

ility, etc., Gaussian type of preference function is suitable

[21]. The type of function used has influence over the precision in WSS. The suitable values of preference crite-ria, threshold and preference/indifference are obtained after conducting experiments multiple times. The indif-ference parameter *‘Q’* is the maximum value of difference function *d(x)* for which alternatives are indifferent. The preference parameter *‘P’* is the minimum value of differ-ence function *d(x)* for which the alternatives are different.

In order to obtain the optimized value of *P* and *Q* param-eter, hardness and satisfaction score are used. The hard-ness of each of the candidate web service is determined using "9,". Eight web services from the dataset of 2507 web services, one from each of the hardness level L3 to L10 are selected. These web services ensure to cover all range of QoS hardness. The optimized value of preference pa-rameter *P* and indifference parameter *Q* is selected for which the satisfaction score is maximized. The satisfac-tion score for ith WS (is evaluated using "10," and "11," [31]. Once the value of *P* and *Q* parameters is ob-tained, the parameter σ which represents point of inflex-ion for Gaussian curve is obtained as mean of *P* and *Q* parameters [6]. Table 6 represents the values of preference (*P*), indifference (*Q*) and σ parameters obtained.

4.2 RESULTS AND EVALUATION

We have performed experimentation using Dataset-1. To further validate the results, we have performed evalua-tion with Dataset-2 as discussed. The results and evalua-tion are presented in this section. Results and evaluation using Dataset-2 is discussed in appendix-A1.

5. DISCUSSION

This paper presents three approaches MSS, HSS and CSS for selection of WSs. MSS and HSS approach is useful for the case when labelled web services are not available for training. HSS approach is preferred in the case when preference of weights of all/partial QoS parameters is available from the end user If no preference of weights is provided, the MSS approach is preferred. From the statis-tical analysis, the performance of MSS and HSS approach is found to be same on satisfaction score. For the case where few labelled WSs are available,the improved WSS results can be obtained using CSS approach. All three approaches uses proposed PROMETHEE Plus algorithm for selection of WSs. The PROMETHEE algorithm is im-proved in three ways – firstly, the end user request is in-cluded with candidate WSs to be processed by PROMETHEE algorithm. Secondly, exact match WS is given preference over the higher match. Thirdly, Gausian and level type preference functions are used during QoS parameters evaluation.

Three proposed approaches are compared by conduct-ing various experiments. The MSS approach has im-proved satisfaction as compared to BPS [7] and EPS [16]. Moreover, the mean Euclidean distance of MSS is lower than BPS. The end user preference of QoS is combined with mathematical model based weight evaluation of MSS approach. This hybrid QoS weight evaluation mech-anism based HSS approach has improved performance and satisfaction over EPS. The classification based prefiltering explores QoS based similarity among web services and classify in the group. The CSS approach has lowest mean Euclidean distance and overall time to select web service. The satisfaction score obtained for CSS ap-proach is observed to be highest and is increasing with the value of *K*. The results and evaluation presented in section 4.2 and appendix-A1 confirms that CSS approach has performed better than existing approaches for WSS. The hardness of query has no/minimum effect on BPS, EPS and HSS. The satisfaction score is observed to be in-creasing for MSS and CSS approach. Therefore, if labelled WSs are available to train the model, then the CSS ap-proach can be employed for web service selection as is also evident from the experimental evaluation. However, if initial set of labelled WSs are not available, then MSS or HSS approach can be preferred for WSS.

6. CONCLUSION

In this paper, a web service selection approach using classification for Prefiltering is proposed. The use of the hybrid weighting scheme brings uniformity in weight calculation and domain dependent inputs are minimized. An improved PROMETHEE method, PROMETHEE Plus, has also been proposed based up-on which the presented selection approach works. Our proposed approach is tested by conducting experi-ments on QWS data set of real-world web services as well as using another dataset generated using available standard dataset generator. The experimental results based on Euclidean distance and satisfaction score show that proposed classification based web service selection approach has ability to find the web services which closely satisfy the end user expectations regard-ing QoS. The satisfaction score from classification based web service selection approach increases with increase in hardness and value of *'K'*. Our proposed approaches for WSS outperforms over other existing PROMETHEE based web service selection approaches.

REFERENCES

[1] S.Y. Hwang, C.C. Hsu, and C.H. Lee, "Service Selection for Web Services with Probabilistic QoS", *IEEE Trans. Services Compu-ting*, Vol. 8, No. 3, pp. 467-480, 2015.

[2] L. Purohit and S. Kumar, "Web Service Selection using Seman-tic Matching", *Proc. ACM Conf. on Advances in Communication Technology & Computing (AICTC '16)*, pp. 16:1 – 16:5, 2016.

[3] S. Kumar and R.B. Mishra, "Cognition based service selection in semantic web service composition", *J. of Computer Science, Infocomp*, Vol. 7, Issue 3, pp. 35-41, 2008.

[4] S. Kumar and N.E. Mastorakis, “Novel Models for Multi-Agent Negotiation based Semantic web service Composition”, *WSEAS Transactions on Computers*, Vol. 9, No. 4, pp. 339-350, 2010.

[5] A. Masri and Q.E. Mahmoud, “QoS-based Discovery and Rank-ing of Web Services”, *Proc. IEEE 16th Conf. on Computer Com-munications and Networks (ICCCN ‘07)*, pp. 529-534, 2007.

[6] J.P. Brans and P. Vincke, “A Preference Ranking Organization METhod”, *Lecture notes in The Institute of Management Sciences*, pp. 647-656, 1985.

[7] C. Herssens, I.J. Jureta, and S. Faulkner, “Dealing with Quality Tradeoffs during Service Selection”, *Proc. IEEE Conf. on Auto-*