Computer Modeling in Engineering & Sciences CMES, vol.123, no.2, pp.525-570, 2020

Toward a More Accurate Web Service Selection Using Modified Interval DEA Models with Undesirable Outputs

Alireza Poordavoodi[1], Mohammad Reza Moazami Goudarzi[2,*], Hamid Haj Seyyed Javadi[3], Amir Masoud Rahmani[4] and Mohammad Izadikhah[5]

Abstract: With the growing number of Web services on the internet, there is a challenge to select the best Web service which can offer more quality-of-service (QoS) values at the lowest price. Another challenge is the uncertainty of QoS values over time due to the unpredictable nature of the internet. In this paper, we modify the interval data envelopment analysis (DEA) models [Wang, Greatbanks and Yang (2005)] for QoS-aware Web service selection considering the uncertainty of QoS attributes in the presence of desirable and undesirable factors. We conduct a set of experiments using a synthesized dataset to show the capabilities of the proposed models. The experimental results show that the correlation between the proposed models and the interval DEA models is significant. Also, the proposed models provide almost robust results and represent more stable behavior than the interval DEA models against QoS variations. Finally, we demonstrate the usefulness of the proposed models for QoS-aware Web service composition. Experimental results indicate that the proposed models significantly improve the fitness of the resultant compositions when they filter out unsatisfactory candidate services for each abstract service in the preprocessing phase. These models help users to select the best possible cloud service considering the dynamic internet environment and they help service providers to improve their Web services in the market.

Keywords: Cloud computing, interval data envelopment analysis, interval entropy, Web service selection, undesirable outputs.

[1] Department of Computer Engineering, Borujerd Branch, Islamic Azad University, Borujerd, Iran.
[2] Department of Mathematics, Borujerd Branch, Islamic Azad University, Borujerd, Iran.
[3] Department of Mathematics and Computer Science, Shahed University, Tehran, Iran.
[4] Department of Computer Science, Khazar University, Baku, Azerbaijan.
[5] Department of Mathematics, Arak Branch, Islamic Azad University, Arak, Iran.
*Corresponding Author: Mohammad Reza Moazami Goudarzi. Email: mr.moazami@iaub.ac.ir.

Received: 18 October 2019; Accepted: 06 February 2020.

1 Introduction

Cloud computing is the widely accepted paradigm of sharing computing resources (e.g., servers, data storage, software and information) as services over the internet. Resources are configurable at the server side, and service providers can manage them with minimum interaction. Therefore, users can access the services rapidly without knowing how they work [Goscinski and Brock (2010); Wei, Vasilakos, Zheng et al. (2010)]. Cloud computing has several technical advantages over traditional computing, such as flexibility, reliability, and availability. Moreover, a business can reduce its expenditures by using metered services (pay-per-use) instead of buying expensive hardware and software. These advantages have led to the rapid development of cloud computing and they encourage large firms to migrate toward cloud services [Goscinski and Brock (2010); Jatoth, Gangadharan and Fiore (2017)].

Cloud computing is based on a service-oriented architecture [Wang, Liu, Sun et al. (2014)], and complex business applications are built by combining atomic Web-based components (i.e., Web services) via the internet [Chen, Dou, Li et al. (2016)].

The nonfunctional properties of Web services are often described by quality-of-service (QoS) attributes such as security, throughput, reliability and response time. The fast growth of Web services has led to the emergence of numerous candidate services with the same functional description but different in QoS values (we refer to the candidate services as concrete services and the functional descriptions as abstract services). As a result, there may be a tradeoff between price and other QoS attributes, which makes it challenging to select the best service for each abstract service [Ouadah, Hadjali, Nader et al. (2019); Karimi, Isazadeh and Rahmani (2017)].

Another issue that should be considered in Web service selection is the emergence of uncertainty and fluctuation in QoS values over different periods. Due to the unpredictable and dynamic internet environment. There are many kinds of uncertainty in the cloud environment, which can affect different QoS attributes. For example, failure and recovery time of servers can affect availability values. Hence, resource migration, data replication, on-demand resource provisioning and server workloads can affect response time values [Tchernykh, Schwiegelsohn, Talbi et al. (2019); Wang, Zheng, Sun et al. (2011); ur Rehman, Hussain and Hussain (2014); Garg, Versteeg and Buyya (2013)]. Hence, the QoS values are not assumed to be exactly obtained. Only their bounded intervals are known, and they are derived from the range of possible values that the corresponding QoS attribute may take.

Cloud services usually consist of several Web services and use graph-based modeling tools to determine the precedence relationships of the user-submitted workflow tasks. In detail, they model a user-submitted request as a directed graph to specify the execution order of the participant Web services [Zhang, Lee and Helal (2019); Wu, Ni, Gu et al. (2010)]. Users require QoS-aware cloud service selection strategies to discover the best concrete service set for a requested workflow that satisfies their QoS constraints. To find the optimal solution to this problem is NP-complete. The problem of cloud service selection usually deals with the following issues: (1) QoS-aware Web service selection: retrieve the worthiest concrete services from UDDI [Zhang, Gong, Lin et al. (2007)] for each abstract service, which can be considered the preprocessing phase for QoS-aware Web service composition approaches. (2) QoS-aware Web service composition: an optimization problem that builds new functionalities (cloud services) by combining different Web services based on the requested workflow. The aggregate QoS of the resultant composition should be maximized possible while satisfying the user's qualitative requests [Huang, Lan and Yang (2009); Jatoth, Gangadharan and Buyya (2019); Zeng, Benatallah, Ngu et al. (2004); Chen, Dou, Li et al. (2016)].

Multi-criteria decision making (MCDM) is a collection of methods to compare and rank multiple options (referred to as alternatives) involving several decision criteria [Özcan, Ünlüsoy and Eren (2017)]. Entropy is a widely accepted method to compute the relative importance of attributes based on their uncertainty in MCDM [Shemshadi, Shirazi, Toreihi et al. (2011)]. Data envelopment analysis (DEA) is a well-established method in the field of MCDM that uses linear programming methodology. This nonparametric method is used to measure the efficiency of a homogeneous group of alternatives (the decision-making unit, or DMU) such as shops, schools, hospitals and cloud services [Karsak (2008); Raju and Kumar (2006); Jatoth, Gangadharan and Fiore (2017)].

In this study, we probe the effects of uncertainty on the internet and propose the most compatible DEA models for mentioned situations. Hence, we can more precisely evaluate the performance of Web services and better identify the qualified concrete services for each abstract service.

As mentioned, only bounded intervals of QoS attributes are known in the dynamic environment of the internet. The traditional DEA models and the entropy method are not suited to imprecise data. Accordingly, we choose the interval DEA models [Wang, Greatbanks and Yang (2005)] and modify them for QoS-aware Web service selection. We construct our models by integrating interval DEA models and interval entropy weights (assuming a uniform distribution of each QoS attribute) [Wu, Sun, Song et al. (2013); Lotfi and Fallahnejad (2010)]. We modify these models based on Russell's model [Pastor, Ruiz and Sirvent (1999)] to improve their accuracy against frequent variations of QoS values. We enrich the models to treat DEA-undesirable (bad) outputs, which jointly are produced along with desirable (good) outputs in the cloud computing environment.

The main conributions of this paper are as follows: (i) We construct a new pair of interval DEA models to rank Web services more precisely. (ii) We propose a solution to discover the best possible cloud service according to the dynamic nature of the cloud environment. For this purpose, we eliminate the unqualified Web services from the design space of metaheuristic algorithms using the proposed models. (iii) We conduct a comprehensive comparison between the proposed models and interval DEA models based on the sensitivity analyses and degree of utilization from the available information to represent the advantages of the proposed models. The rest of this paper is organized as follows. In the next section, related work and the role of MCDM in cloud service selection are presented. The concepts of DEA, interval DEA, entropy, interval entropy, and comparision of interval efficiency scores are reviewed in Section 3. The proposed models for cloud service selection are constructed in Section 4. The usefulness of proposed models for QoS-aware Web service composition is discussed in Section 5. In Section 6, an experiment is conducted to describe Web service selection using the proposed models and a synthesized dataset. In Section 7, the capabilities of the proposed models are analyzed. In Section 8, the experimental results are presented when proposed models are employed in the preprocessing phase of GA. Conclusions are given in Section 9.

2 Related work

2.1 Web and cloud service selection

So far, a lot of researchers have gone for Web and cloud service selection. In this section, we will present a brief overview of MCDM for these problems.

Ouadah et al. [Ouadah, Hadjali, Nader et al. (2019)] proposed SEFAP, a hybrid MCDM method to rank Web services. They combined fuzzy AHP and Entropy for weighting selection criteria, and they used PROMETHEE to rank Web services. Serrai et al. [Serrai, Abdelli, Mokdad et al. (2017)] proposed a hybrid method which consists of BWM and some other MCDM methods to rank Web services. Serrai et al. [Serrai, Abdelli, Mokdad et al. (2019)] developed a Web service selection approach to cope with user constraints on QoS criteria. For this purpose, they extended the TOPSIS, SAW, WPM and VIKOR methods to a normalization technique called OMRI. They also considered an extension of the AHP method to weight QoS criteria. Sun et al. [Sun, Zhang and Liu (2016)] presented an MCDM method for service ranking. They integrated the entropy and subjective weights to determine the importance of QoS attributes. Then they ranked services by the TOPSIS method based on their quality.

The following studies concentrated only on a single task in the cloud environment and used MCDM for cloud service selection.

Al-Faifi et al. [Al-Faifi, Song, Hassan et al. (2019)] developed a hybrid MCDM method to evaluate and rank cloud services from smart data. Their method consists of two components: (i) clustering the cloud services using the k-means algorithm and (ii) ranking the obtained clusters using MCDM methods to make a final decision. Ma et al. [Ma, Hu, Li et al. (2019)] proposed a variation-aware cloud service selection via collaborative QoS prediction to select an optimal cloud service according to the user's non-functional requirements. They employed cloud model theory to compute the uncertainty of QoS attributes and ranked cloud services based on both user preferences and QoS variation using an improved TOPSIS. Gireesha et al. [Gireesha, Somu, Krithivasan et al. (2020)] presented a novel approach for cloud service provider selection based on their trustworthy. They also employed a weight assessment method to determine the importance of QoS attributes based on objective and subjective measures. Hussain et al. [Hussain, Chun and Khan (2020)] proposed a novel framework called CSSaaS, which provides essential facilities for viable cloud service selection as a service. Then, they introduced a novel MCDM approach named FLBWN for viable cloud service selection of CSSaaS framework under a fuzzy environment. Gireesha et al. [Gireesha, Somu, Raman et al. (2020)] utilized the neural networks as an MCDM method to solve the cloud service selection problem. In detail, they determined the optimal weights of QoS attributes using WNN. Then, they employed EDAS to rank cloud service providers. Silas et al. [Silas, Rajsingh and Ezra (2012)] presented a cloud service selection middleware named SSM_EC which used ELECTERE methodology. Rai et al. [Rai and Kumar (2016)] developed a new method using TOPSIS and VIKOR for IaaS cloud service selection. This method ranks cloud services daily. To obtain more accurate results, it assigns more weight to recent values. Jatoth et al. [Jatoth, Gangadharan and Fiore (2017)] developed new DEA models integrated with AHP/ANP to evaluate cloud services based on user preferences. Rehman et al. [ur Rehman, Hussain and Hussain (2014)] presented an MCDM method for IaaS cloud service selection considering fluctuations of services' efficiency. This method performs parallel MCDM according to user preferences over different periods of time. Then, it determines the overall best service by aggregating the individual results of each time period. Sun et al. [Sun, Ma, Zhang et al. (2016)] proposed an MCDM method based on the fuzzy AHP and fuzzy TOPSIS approaches to rank cloud services in uncertain cloud environments, and built a fuzzy ontology model to filter the services by function matching before ranking them. Kumar et al. [Kumar, Mishra and Kumar (2017)] devised an integrated MCDM method for cloud service selection in a fuzzy environment. Their method determines the weights of QoS attributes using AHP and ranks cloud services using fuzzy TOPSIS. Jatoth et al. [Jatoth, Gangadharan, Fiore et al. (2019)] proposed an MCDM method derived from Grey TOPSIS and AHP for cloud service selection. Xu et al. [Xu, Ma and Wang (2015)] utilized CCR and BCC DEA models to classify cloud services by efficiency levels, and provided guidelines to improve less efficient services. Azadi et al. [Azadi, Emrouznejad, Ramezani et al. (2019)] proposed a network DEA model to evaluate the efficiency of cloud services more accurately. Moreover, their model can detect inefficient divisions of cloud services. This is a key advantage of their approach over all the other existing approaches. Shetty et al. [Shetty and DMello (2015)] eliminated services based on users' QoS constraints ranked qualified services using the REMBRANDT approach. Kumar et al. [Kumar and Kumar (2017)] proposed a hybrid method for cloud service selection. They used entropy to determine the weight of each QoS attribute and TOPSIS to evaluate the efficiency of cloud services.

An important factor considered in some of above studies is uncertainty and fluctuation of QoS values in the dynamic internet environment [ur Rehman, Hussain and Hussain (2014); Sun, Ma, Zhang et al. (2016); Kumar, Mishra and Kumar (2017); Wang, Zheng, Sun et al. (2011); Kumar and Kumar (2017); Ma, Hu, Li et al. (2019)]. For example, Wang et al. [Wang, Zheng, Sun et al. (2011)] used a cloud model to determine QoS uncertainty to eliminate the unqualified Web services. Based on QoS uncertainty, they selected the optimal Web service.

All of the mentioned studies select the best possible cloud or Web services using different MCDM methods. A part of these approaches which considers the uncertainty factor in their selection process has merely concentrated on a single task. But the current business processes are usually complicated and involve several tasks. Hence, an unsupported factor that needs to be investigated in such business processes is the ability of the ranking method to select the best possible cloud service considering the variations of QoS values. In this study, we aim at presenting a hybrid MCDM method to fill this gap. The conspicuous components of this method are: (i) a pair of modified interval DEA models to select best Web services considering the uncertainty of the internet. (ii) a Web service composition approach to discover the best possible cloud service among the selected Web services considering the uncertainty of the internet.

2.2 Problem definition and MCDM in cloud service selection

In many real-world problems, we face situations in which the best alternative should be selected from several options. In this case, an alternative may be best based on one or more criteria and worse on other criteria. Hence, a compromise is required to rank the alternatives.

MCDM is a sub-discipline of operations research used rank alternatives by aggregating multiple conflicting/related criteria. Hence, MCDM includes appropriate methods to solve the above problem, and it is widely employed in decision support systems [Pérez, Laprise and Rey (2018); Kwok and Lau (2019), Hasnain, Thaheem and Ullah (2018); Eraslan and Ic (2019); Santos, Bressi, Cerezo et al. (2019)]. ELECTRE, TOPSIS, VIKOR, AHP, ANP, and DEA are examples of well-known MCDM methods. A variety of studies, which we briefly reviewed in Section 2.1, have compared Web services using MCDM methods.

In the QoS-aware Web service selection problem, the best service is selected according to its QoS attributes. Hence, this problem falls into the category of MCDM methods.

In this paper, we present interval DEA models that are customized for the dynamic internet environment. These models can help users select the best Web service more accurately according to the circumstances of the cloud environment.

To understand the application of the proposed models in Web service composition problem, consider the AgFlow middleware platform [Zeng, Benatallah, Ngu et al. (2004)]. AgFlow is used for Web service composition, and its architecture is depicted in Fig. 1.

In this architecture, service providers register the QoS and other descriptions of their Web services in the UDDI registry. When the composition manager receives a request, an instance of composite service is initiated and a service broker is invoked to retrieve appropriate candidate services for each abstract service. Thereafter, it aggregates retrieved Web services and returns the optimal service set based on the requested workflow. In this architecture, proposed models will be embedded in the service broker and they will employ Web service selection considering the uncertainty of the cloud environment.

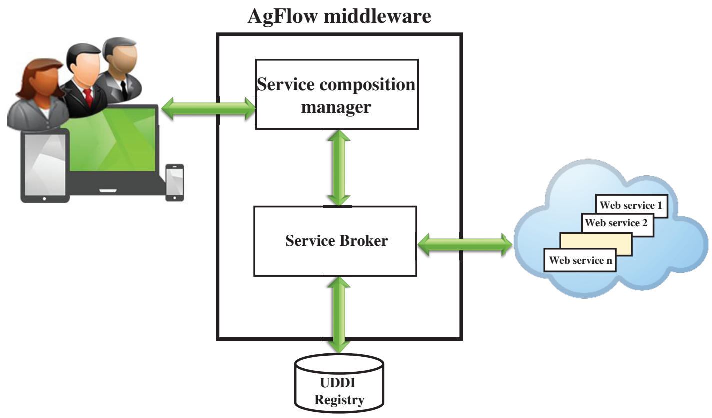


Figure 1: AgFlow's architecture

3 Preliminaries

This section introduces the concepts of DEA, interval DEA, entropy and interval entropy and their roles in evaluation of Web services.

3.1 DEA

As mentioned in Section 1, DEA is a method based on MCDM to evaluate the relative efficiency of a group of homogeneous DMUs. DMUs represent processing units that convert several inputs to several outputs. This method was first proposed by Charnes et al. [Charnes, Cooper and Rhodes (1978)]. It has gradually been expanded to a variety of applications, and is applied to many fields including engineering, economics, and management. In DEA, the production possibility set (PPS) [Cooper, Seiford and Tone (2007)] and the corresponding models are constructed by the acceptance of a series of assumptions from a set of observed and virtual (possible) DMUs (A virtual DMU is an ideal DMU, which can be made by combining a set of observed DMUs. Since this DMU does not necessarily exist, it sometimes called a virtual DMU). After construction of PPS, DEA determines the "efficient frontier" which includes all efficient DMUs. The more inefficient DMU the further it is from the efficient frontier. There are two conventional methods to determine the efficiency score: (1) the input-oriented method, which is to move toward the efficient frontier by reducing inputs while maintaining the same level of outputs; and (2) the output oriented method, which is to increase outputs while keeping inputs fixed at their current values [Banker, Charnes and Cooper (1984), Lotfi, Jahanshahloo, Ebrahimnejad et al. (2010); Wu, Sun, Song et al. (2013); He, Zhang, Lei et al. (2013)].

We use an output-oriented CCR model [Lotfi, Jahanshahloo, Ebrahimnejad et al. (2010)] whose envelopment form is presented in model (1).

$$\varphi^* = \max \varphi$$
$$s.t \begin{cases} \sum_{j=1}^{n} \lambda_j x_{ij} \leq x_{id} & i = 1, 2, \ldots, m \\ \sum_{j=1}^{n} \lambda_j y_{rj} \geq \varphi y_{rd} & r = 1, 2, \ldots, s \\ \lambda_j \geq 0 & j = 1, 2, \ldots, n \end{cases} \quad (1)$$

In model (1), x_{ij} and y_{rj} respectively represent the ith input and rth output of DMU_j where i=(1, 2, …, m), r=(1, 2, …, s), and j=(1, 2, …, n). The DMU under evaluation is DMU_d. The variables of this model are λ and φ. If φ*=1, then DMU_d is efficient. Otherwise, if φ*>1, then this DMU is inefficient, and it is possible to increase outputs with keeping inputs unchanged. The efficiency score of DMU_d is $\frac{1}{\varphi^*}$. The variable λ is a weight vector that indicates the contribution of DMU_j in finding the best virtual DMU for DMU_d, where j=(1, 2, …, n).

We aim to customize this basic DEA model for Web service evaluation according to the dynamic internet environment. For this purpose, we consider a Web service as a DMU, QoS attributes as outputs and price items (the cost of service for a request) as inputs, as shown in Fig. 2.

Figure 2: DEA model for Web service evaluation

3.2 Interval DEA

Traditional DEA models suppose that DMU data are deterministic, whereas due to data uncertainty of Web services, the input and output data cannot be determined precisely [Jahanshahloo, Lotfi and Davoodi (2009)]. In such circumstances, the ith input and rth output of DMU_j are represented by $[x_{ij}^l, x_{ij}^u]$ and $[y_{rj}^l, y_{rj}^u]$, respectively where x_{ij}^l>0 and y_{rj}^l>0. To address such circumstances, Wang et al. [Wang, Greatbanks and Yang (2005)] proposed interval DEA models, which were supported by other researchers [Jahed, Amirteimoori and Azizi (2015), Wu, Sun, Song et al. (2013)]. Models (2) and (3) are their envelopment form [Cooper, Seiford and Tone (2007)], which compute the lower and upper bounds of φ*.

$$\varphi^{l*} = \max \varphi$$
$$s.t \begin{cases} \sum_{j=1}^{n} \lambda_j x_{ij}^l \leq x_{id}^l & i=1,2,\ldots,m \\ \sum_{j=1}^{n} \lambda_j y_{rj}^u \geq \varphi y_{rd}^u & r=1,2,\ldots,s \\ \lambda_j \geq 0 & j=1,2,\ldots,n \end{cases} \tag{2}$$

$$\varphi^{u*} = \max \varphi$$
$$s.t \begin{cases} \sum_{j=1}^{n} \lambda_j x_{ij}^l \leq x_{id}^u & i=1,2,\ldots,m \\ \sum_{j=1}^{n} \lambda_j y_{rj}^u \geq \varphi y_{rd}^l & r=1,2,\ldots,s \\ \lambda_j \geq 0 & j=1,2,\ldots,n \end{cases} \tag{3}$$

In models (2) and (3), x_{ij}^l and x_{ij}^u represent the lower and upper bounds of the ith input of DMU_j where i=(1, 2, …, m) and j=(1, 2, …, n). Also, y_{rj}^l and y_{rj}^u represent the lower and upper bounds of the rth output of DMU_j where r=(1, 2, …, s) and j=(1, 2, …, n). Morever, we always have: $0<x_{ij}^l \leq x_{ij}^u$ and $0<y_{rj}^l \leq y_{rj}^u$. The other notations of models (2) and (3) (λ and φ) have their usual meanings, as defined in model (1). Models (2) and (3) calculate the efficiency of DMUs in optimistic and pessimistic viewpoints, respectively. In detail, model (2) use the best characteristics of DMU_d (x_{id}^l and y_{rd}^u) in its calculation and model (3) use the worst ones (x_{id}^u and y_{rd}^l). In these models, we always have: $\varphi^l{*} \leq \varphi^u{*}$. Since these models are output-oriented, the efficiency score of DMU_d is computed by $\left[\frac{1}{\varphi^{u*}}, \frac{1}{\varphi^{l*}}\right]$. DMU_d is efficient from the optimistic viewpoint if $\varphi^l{*}=1$. Otherwise, DMU_d is inefficient.

3.3 Comparing interval efficiency scores

After determining the relative efficiency of Web services using models (2) and (3), we must rank them to find the best. Thus the interval numbers should be compared. Various methods exist to compare interval numbers according to different theories [Sengupta and Pal (2000); Chanas and Zielinski (1999); Li, Zeng and Yin (2018), Ramón, Ruiz and Sirvent (2014); Wang, Chin and Yang (2007)]. In this research, we use the following two approaches:

1. Geometric average [Wang, Chin and Yang (2007)]: The interval numbers are compared based on their midpoints. The midpoint of the efficiency scores is computed as the second root of the products of $\varphi^l{*}$ and $\varphi^u{*}$. Since the models (2) and (3) are output-oriented, we must reverse their results and then compute the overall efficiency score of the jth Web service using Eq. (4).

$$Geom_j = \sqrt[2]{\left(\frac{1}{\varphi_j^{l*}}\right) \times \left(\frac{1}{\varphi_j^{u*}}\right)}, \quad j = 1, 2, \ldots, n \tag{4}$$

2. Sengupta's approach [Sengupta and Pal (2000)]: compares the interval numbers A and B based on their midpoint and half-with. for which we define an acceptability function using Eq. (5).

$$\mathcal{A}_{\oplus}(A, B) = \frac{m(B) - m(A)}{w(B) + w(A)} \tag{5}$$

In Eq. (5), Assume $m(A)$ and $w(A)$ are the midpoint and half-with, respectively, of A, and $m(B)$, $w(B)$ are the midpoint and half-with, respectively, of B. The value of $A_{\oplus}(A,B)$ represents the grade of satisfaction of the premise "A is less than B". Furthermore, if A and B have the same mid-point ($m(A)$=$m(B)$), then this function selects the number with less uncertainty.

3.4 Entropy

Entropy is an objective weighting method that was proposed by Shannon [Shannon (1948)]. This method is used in various fields, such as information theory, physics, and transportation models.

It measures the amount of information that each attribute provides for decision makers. If one attribute contains more information than another, then it has more discrimination power in decision making. For this reason, this method assigns more weight to it. Thus, the results of this method can be used as the relative importance (weight) of QoS attributes in MCDM [Lotfi and Fallahnejad (2010); ur Rehman, Hussain and Hussain (2014)]. Fig. 3 shows the decision matrix for weighing QoS attributes of Web services.

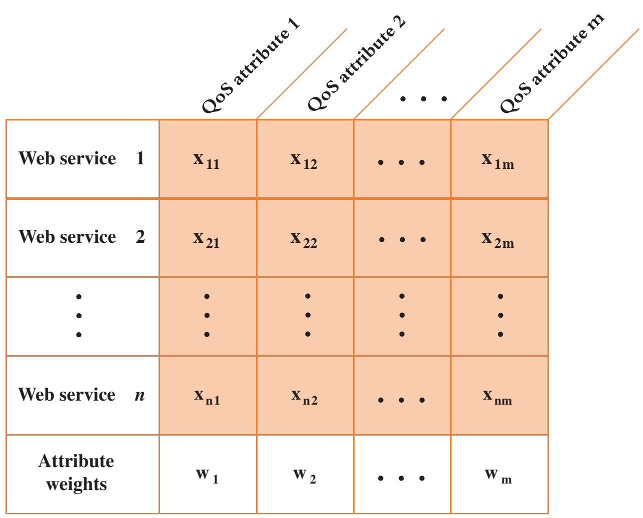


Figure 3: Decision matrix for weighing to the QoS attributes

Weights of Shannon's entropy are computed as follows. First, the decision matrix elements are normalized using Eq. (6).

$$p_{ij}=\frac{x_{ij}}{\sum_{j=1}^{n} x_{ij}}, \qquad j=1,\ldots,n, \qquad i=1,\ldots,m \tag{6}$$

In Eq. (6), x_{ij} represents the value of the jth attribute with respect to the ith alternative in the decision matrix. The variables m and n represent the numbers of attributes and alternatives, respectively. Then, the entropy h_i is computed by Eq. (7).

$$h_i=-k\sum_{j=1}^{n} p_{ij}.\ln p_{ij}, \quad i=1,\ldots,m \tag{7}$$

Where k is the entropy constant and is equal to $(\ln \quad n)^{-1}$, and $p_{ij}.ln\ p_{ij}$ is defined as 0 if p_{ij}=0. The weight of the ith attribute is computed using Eq. (8).

$$w_i=\frac{1-h_i}{m-\sum_{s=1}^{m} h_i}, \quad i=1,..,m \tag{8}$$

3.5 Interval entropy

Due to the uncertainty of the cloud environment, each element of the decision matrix (Fig. 3) can be considered an interval number (a continuous probabilistic distribution is assumed for each QoS attribute). For example, the lower and upper bounds of the jth QoS attribute with respect to the ith Web service is represented by $[x_{ij}^{l}, x_{ij}^{u}]$. Under this condition, the weight of the QoS attributes should be modified to the interval form to obtain the correct result. For instance, the weight of the ith QoS attribute should be changed to $[w_i^l, w_i^u]$. Lotfi et al. [Lotfi and Fallahnejad (2010)] proposed an interval entropy method to deal with this problem. First, they normalized the decision matrix elements using Eq. (9).

$$p_{ij}^{l}=\frac{x_{ij}^{l}}{\sum_{j=1}^{n} x_{ij}^{u}}, \quad p_{ij}^{u}=\frac{x_{ij}^{u}}{\sum_{j=1}^{n} x_{ij}^{u}}, \qquad i=1,\ldots,m, \qquad j=1,\ldots,n \tag{9}$$

Then, they computed the lower and upper bounds of interval entropy using Eqs. (10) and (11), respectively.

$$h_i^l=\min\left\{-k\sum_{j=1}^{n} p_{ij}^{l}.\ln p_{ij}^{l}, -k\sum_{j=1}^{n} p_{ij}^{u}.\ln p_{ij}^{u}\right\}, \quad i=1,\ldots,m \tag{10}$$

$$h_i^u=\max\left\{-k\sum_{j=1}^{n} p_{ij}^{l}.\ln p_{ij}^{l}, -k\sum_{j=1}^{n} p_{ij}^{u}.\ln p_{ij}^{u}\right\}, \quad i=1,\ldots,m \tag{11}$$

where k is the entropy constant and is equal to $(\ln \quad n)^{-1}$, and $p_{ij}^{l}.\ln p_{ij}^{l}$ is defined 0 if p_{ij}^{l}=0 and $p_{ij}^{u}.\ln p_{ij}^{u}$ is defined 0 if p_{ij}^{u}=0. The lower and upper bounds of the weights of the ith

QoS attribute are computed using Eqs. (12) and (13), respectively.

$$w_i^l = \frac{1 - h_i^u}{m - \sum_{s=1}^{m} h_s^l}, \quad i = 1, .., m \tag{12}$$

$$w_i^u = \frac{1 - h_i^l}{m - \sum_{s=1}^{m} h_s^u}, \quad i = 1, .., m \tag{13}$$

When all of the Web services have deterministic QoS values, the the interval entropy weights lead to the usual entropy weight. But if at least one of the QoS values is in the interval form, all weights will be in the interval form too.

4 QoS-aware Web service selection using proposed models

The main objective of this study is to select the best Web service based on the characteristics of the cloud environment. To deal with this problem, we use the interval entropy to determine the weights (priorities) of different QoS attributes of Web services. Then, we integrate the interval entropy weights and interval DEA models to select the best Web service. We modify the proposed models based on Russell's model, since the interval DEA models cannot capture all of the QoS variations in their evaluations. Finally, we extend the proposed models to treat undesirable (bad) QoS attributes in the cloud environment along with desirable (good) ones.

Fig. 4 describes the process of QoS-aware Web service selection using the proposed models. The UDDI registry, which is used as an input layer, includes the list of concrete services and their QoS values. The main operations of this process shown in the white boxes. The inputs and outputs are represented with blue boxes. The workflow of these operations is as follows.

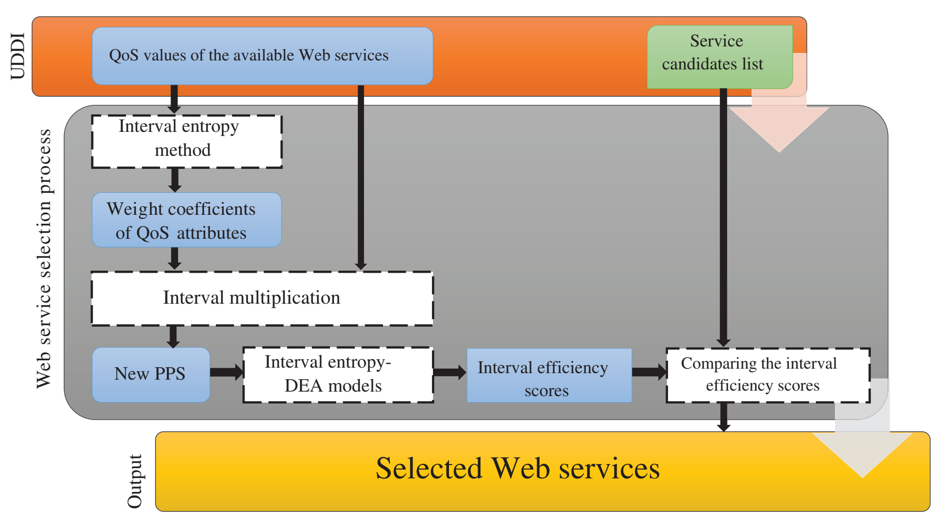


Figure 4: The process of QoS-aware Web service selection using proposed models

First, the interval entropy method is used to determine the interval weights of inputs and outputs. Figs. 5(a) and 5(b) show the decision matrices for weighing the lower and upper bounds of the QoS attributes, respectively. In these figures, x^l_{ij} represents the lower bound of QoS values where i=(1, 2, …, n) and j=(1, 2, …, m). Similarly, x^u_{ij} represents the upper bound of them.

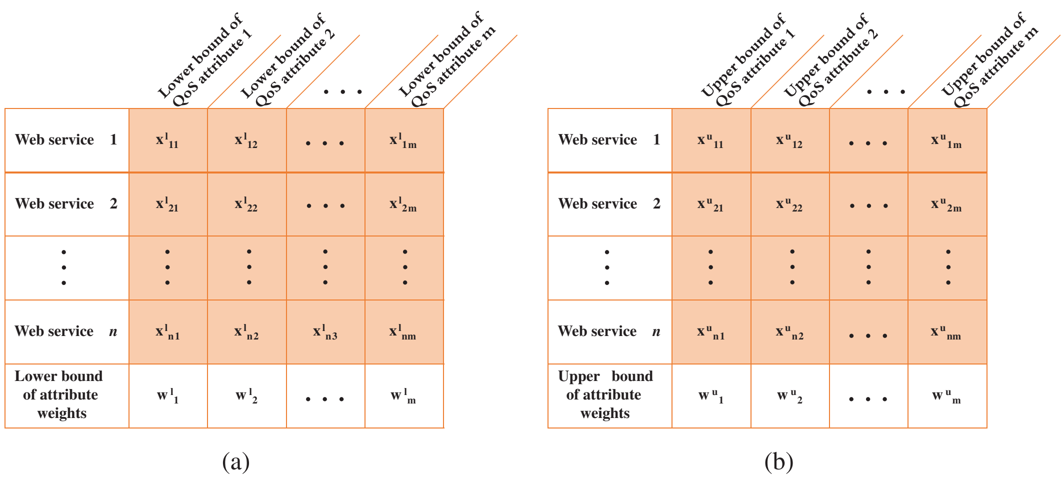


Figure 5: Decision matrices for weighing the lower and upper bounds of the QoS attributes

Then a new PPS is constructed, which includes the interval entropy weights. For this purpose, each interval weight is multiplied by the interval value of the corresponding input/output using interval multiplication [Chakraverty (2014)]. For example, $[y^{l'}_{rj}, y^{u'}_{rj}]$ represents the rth output of DMU_j, which is computed by Eq. (14), where r=(1, …, s) and j=(1, …, n).

$$\begin{aligned}[y^{l'}_{rj}, y^{u'}_{rj}] &= [y^l_{rj}, y^u_{rj}] \times [w^l_r, w^u_r] \\ &= [\min\{y^l_{rj}w^l_r, y^u_{rj}w^l_r, y^l_{rj}w^u_r, y^u_{rj}w^u_r\}, \\ &\quad \max\{y^l_{rj}w^l_r, y^u_{rj}w^l_r, y^l_{rj}w^u_r, y^u_{rj}w^u_r\}]\end{aligned} \tag{14}$$

The proposed models are constructed in the third step. These are run separately for each Web service to compute their interval efficiency scores. (This is discussed further in Section 4.1).

Finally, the best Web services are selected by comparing interval efficiency scores.

These steps are repeated for each abstract service to determine the qualified Web services.

4.1 Proposed models

In this section, we construct our DEA models to compute the relative efficiency of Web services considering the uncertainty of the cloud environment. Then we rank the Web services based on their efficiency scores. To rank Web services using DEA, they should be grouped by functional matchmaking in the UDDI as shown in Fig. 6. As is seen, Web services of each group have equal numbers of inputs and outputs. For example, each candidate service of abstract service 1 has s QoS attributes and m price items.

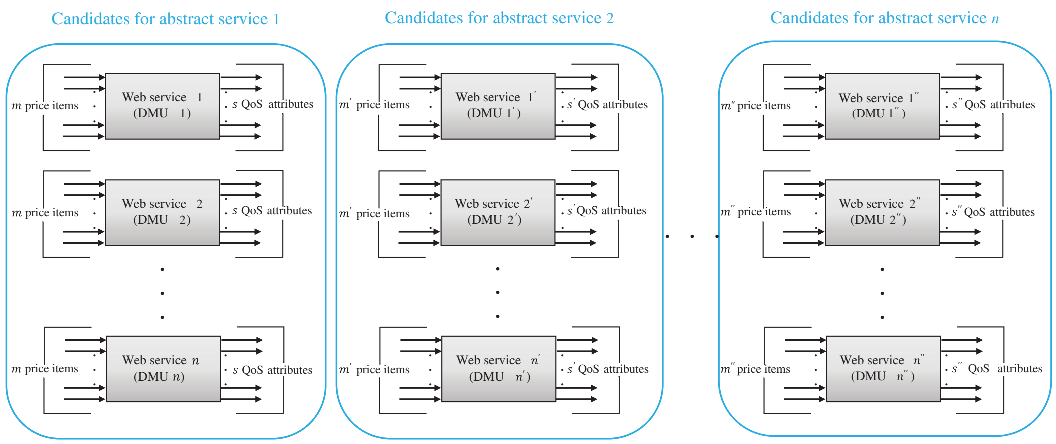


Figure 6: DEA model of UDDI

As mentioned, we consider inputs and outputs of Web services as continuous interval variables due to the dynamic nature of the internet. Hence, we construct the proposed models based on the interval DEA models described in Section 3.2.

In Section 3.1, we introduced the output-oriented CCR model to evaluate Web services. This model is constructed based on the constant return to scale (CRS) hypothesis, in which input and output variables change proportionally. For example, if the input values for a DMU are all doubled, then the DMU must produce twice as many outputs. CRS in the cloud environment is not acceptable due to internet dynamics and different pricing plans of service level agreements (SLAs) [Serrano, Kouki and Ledoux (2016)]. Hence, we use the BCC model which assumes a variable return to scale (VRS). In this hypothesis, the input and output variables can change disproportionately with each other. The BCC model includes the constraints of the CCR model and $\sum_{j=1}^{n} \lambda_j=1$ as an additional constraint. This constraint denotes the assumption of the VRS [Cooper, Seiford and Tone (2007)].

Proposed models are provided for use in the service broker. These models should run periodically to precisely reflect the intrinsic changes in QoS values [Jula, Sundararajan and Othman (2014)]. As each QoS attribute corresponds to a specific aspect of a Web service's efficiency (e.g., reliability, security, and throughput), it does not reflect the reality that all of the outputs of DMUs increase at the same rate. QoS attributes should be increased at different ratios, while interval DEA models can only decrease/increase all of the inputs/outputs at the same rate (because models (2) and (3) use a unique φ variable for all the outputs). Therefore, if we use the interval DEA models to rank Web services in the service broker, they cannot consider all of the QoS variations in their evaluations. (This is addressed in Section 7.3 using a synthesized dataset.) To remove this drawback, the integrated interval entropy-DEA models are presented based on Russell's model as follows.

$$\gamma^{l*} = \max \frac{1}{s}\sum_{r=1}^{s}\gamma_r$$
$$s.t\begin{cases}\sum_{j=1}^{n}\lambda_j x_{ij}^{l'} \le x_{id}^{l'} & i=1,2,\ldots,m\\ \sum_{j=1}^{n}\lambda_j y_{rj}^{u'} \ge \gamma_r y_{rd}^{u'} & r=1,2,\ldots,s\\ \sum_{j=1}^{n}\lambda_j = 1 & \\ \lambda_j \ge 0 & j=1,2,\ldots,n\\ \gamma_r \ge 1 & r=1,2,\ldots,s\end{cases} \tag{15}$$

$$\gamma^{u*} = \max \frac{1}{s}\sum_{r=1}^{s}\gamma_r$$
$$s.t\begin{cases}\sum_{j=1}^{n}\lambda_j x_{ij}^{l'} \le x_{id}^{u'} & i=1,2,\ldots,m\\ \sum_{j=1}^{n}\lambda_j y_{rj}^{u'} \ge \gamma_r y_{rd}^{l'} & r=1,2,\ldots,s\\ \sum_{j=1}^{n}\lambda_j = 1 & \\ \lambda_j \ge 0 & j=1,2,\ldots,n\\ \gamma_r \ge 1 & r=1,2,\ldots,s\end{cases} \tag{16}$$

In models (15) and (16), γ_r is the efficient value of the rth output and the objective function is defined as the average of efficient values. Hence, these models can change outputs at different rates, which is more compatible with the dynamic situation of the cloud environment. Other notations of the mentioned models were defined previously. As mentioned in Section 3.2, the efficiency score of DMU_d is computed through $[\frac{1}{\gamma^{u*}}, \frac{1}{\gamma^{l*}}]$.

In models (15) and (16), DMU_d is efficient if γ_r^*=1 for r=(1, 2, .., s). If we remove the constraint $\gamma_r \ge 1$, we may have $\gamma_r^* < 1$ for some outputs of an optimal solution. This inequality means a kind of reduction in outputs, which is inconsistent with the concept of output-oriented models. Consequently, we consider the constraint $\gamma_r \ge 1$ for r=(1, 2, .., s) in models (15) and (16).

Definition 4.1 In model (15), when $\gamma_r \ge 1$ for $r=(1, 2, .., s)$, DMU$_d$ is efficient from the optimistic viewpoint if γ^{l*}=1. Otherwise, DMU$_d$ is inefficient and we have $\gamma^{l*} > 1$.

4.2 Proposed models considering undesirable outputs

Sometimes undesirable (bad) inputs and outputs may jointly appear along with desirable (good) inputs and outputs. Traditional DEA models only values the desirable and simply ignores the undesirable. In their presence, a DMU's efficiency improves by decreasing undesirable outputs and desirable inputs and increasing undesirable inputs and desirable outputs.

As mentioned, we consider QoS attributes as outputs of the proposed models. Hence, both desirable and undesirable outputs may exist in the proposed models. For example, consider the following QoS attributes:

- Availability: the percentage of time in which a service is operating and accessible upon request.
- Throughput: the number of tasks that can be completed by a service provider at a given time.
- Reliability: the probability that a service operates during a given time without failure, based on the promise of the service provider and past failures experienced by users.
- Adaptability: a service provider's ability to modify its service's functionality based on the requests of users, defined as the time needed to adapt a service to the requests.
- Response time: the period of time between sending a request to a Web service and receiving its response [Zheng, Wu, Zhang et al. (2012); Garg, Versteeg and Buyya (2013)].

According to the above definitions, we observe that throughput, reliability, and availability are desirable outputs (greater values are favorable), while response time and adaptability are undesirable (smaller values are favorable). We can conclude that Web services produce both desirable and undesirable outputs, in which case traditional DEA models are not applicable. We evaluate QoS attributes independently [Zheng, Wu, Zhang et al. (2012)]. Accordingly, we can suppose strong disposability of undesirable QoS attributes. Hence, we treat undesirable outputs using the approach developed by Seiford et al. [Seiford and Zhu (2002)] and supported by other researchers [Anvari, Zulkifli, Sorooshian et al. (2014); Liu, Chu, Yin et al. (2017)]. This approach is appropriate for our purpose since it preserves the convexity and linearity of the BCC model. We extend this approach to interval form for adaptation with the interval DEA models.

We first use Eqs. (17) and (18), respectively, to transform the lower and upper bounds of undesirable outputs.

$$\bar{y}_p^{bu} = -y_p^{bl} + w_p > 0, \quad (p = 1, 2, \ldots, k) \tag{17}$$

$$\bar{y}_p^{bl} = -y_p^{bu} + w_p > 0, \quad (p = 1, 2, \ldots, k) \tag{18}$$

In Eqs. (17) and (18), ${y^{bl}}_p$ and ${y^{bu}}_p$ are the lower and upper bounds, respectively, of undesirable outputs, and w_p is a proper translation vector which transforms the lower and upper bounds of the negative undesirable outputs to positive ones where p=(1, 2, …, k). After applying these transformations, we can use the undesirable outputs in our proposed models. Thus models (15) and (16) turn into models (19) and (20).

$$\gamma^{l*}=\max\frac{1}{s}\sum_{r=1}^{s}\gamma_r$$
$$s.t\begin{cases}\sum_{j=1}^{n}\lambda_j x_{ij}^{l'}\leq x_{id}^{l'} & i=1,2,\ldots,m\\ \sum_{j=1}^{n}\lambda_j y_{qj}^{gu'}\geq\gamma_q y_{qd}^{gu'} & q=1,2,\ldots,t\\ \sum_{j=1}^{n}\lambda_j \bar{y}_{pj}^{bu'}\geq\gamma_p \bar{y}_{pd}^{bu'} & p=1,2,\ldots,k\\ \sum_{j=1}^{n}\lambda_j=1 & \\ \lambda_j\geq 0 & j=1,2,\ldots,n\\ \gamma_r\geq 1 & r=1,2,\ldots,s\end{cases}\tag{19}$$

$$\gamma^{u*}=\max\frac{1}{s}\sum_{r=1}^{s}\gamma_r$$
$$s.t\begin{cases}\sum_{j=1}^{n}\lambda_j x_{ij}^{l'}\leq x_{id}^{u'} & i=1,2,\ldots,m\\ \sum_{j=1}^{n}\lambda_j y_{qj}^{gu'}\geq\gamma_q y_{qd}^{gl'} & q=1,2,\ldots,t\\ \sum_{j=1}^{n}\lambda_j \bar{y}_{pj}^{bu'}\geq\gamma_p \bar{y}_{pd}^{bl'} & p=1,2,\ldots,k\\ \sum_{j=1}^{n}\lambda_j=1 & \\ \lambda_j\geq 0 & j=1,2,\ldots,n\\ \gamma_r\geq 1 & r=1,2,\ldots,s\end{cases}\tag{20}$$

In models (19) and (20), ${y^{gl}}_q$ and ${y^{gu}}_q$ are the lower and upper bounds, respectively, of desirable outputs, where q=(1, 2, …, t) and k+t=s. Other notations were previously defined.

DMU$_d$ is efficient from the optimistic viewpoint if the equality γ^{l}*=1 hold [Wang, Greatbanks and Yang (2005)]. No improvement is possible in the outputs of the Web service because we have an optimal solution: γ_r^*=1 for r=(1, …, s).

Here, we compute the proportion of desirable/undesirable outputs in an inefficient Web service, which should be increase/decrease to make it efficient. Model (20) uses the pessimistic perspective. Therefore, the suggested improvements of this model are redundant and unnecessary in most situations. Hence, we utilize model (19) to identify the necessary improvements, which can apply to the service's outputs in any situation. In this model, it is necessary to increase/decrease the upper/lower bound of the rth desirable/undesirable output of DMU_d at rate γ_r^*-1, where $r=(1, 2, \ldots, s)$, with a fixed input to turn it into an efficient service if $\gamma_r^*>1$. In fact, the above rate play the role of auxiliary variable in the slack-based models [Tone (2001)]. In Theorem 4.1, we observe an important property of the proposed models, which shows that our models are always feasible.

Theorem 4.1 Models (19) and (20) are feasible.

Proof. set

$$\begin{array}{llll} \lambda_d=1, & \lambda_j=0, & j=1,\ldots,n, & j\neq d \\ \gamma_r=1, & r=1,\ldots,s & & \end{array}$$

It can be seen that (γ_r, λ_j) with the above components is a feasible solution for these models. ▪

Another property that is considered after the construction of DEA models is their boundedness. In Theorem 4.2, we prove that the proposed models are bounded.

Theorem 4.2 Models (19) and (20) are bounded.

Proof. The dual (multiplier) form of the model (19) is expressed as the model (21).

$$\min Z=\sum_{r=1}^{s}-\beta_r+\sum_{i=1}^{m}v_i x_{id}^{l'}+u_0$$
$$s.t\begin{cases} \sum_{i=1}^{m} v_i x_{ij}^{l'}-\sum_{q=1}^{t}\mu_q y_{qj}^{gu'}-\sum_{p=1}^{k}\omega_p \bar{y}_{pj}^{bu'}+u_0\geq 0 & j=1,2,\ldots,n \\ \mu_q y_{qd}^{gu'}-\beta_q=\dfrac{1}{s} & q=1,2,\ldots,t \\ \omega_p \bar{y}_{pd}^{bu'}-\beta_p=\dfrac{1}{s} & p=1,2,\ldots,k \\ \mu_q\geq 0 & q=1,2,\ldots,t \\ \omega_p\geq 0 & p=1,2,\ldots,k \\ v_i\geq 0 & i=1,2,\ldots,m \\ \beta_r\geq 0 & r=1,2,\ldots,s \\ u_0 & free \end{cases} \tag{21}$$

It can be seen that the solution $(\beta, \mu, \omega, v, u_0)$ with the following components are a feasible solution for model (21). Therefore, model (19) is bounded based on the weak duality theorem [Vazirani (2013)].

$$\beta_r = 0 \qquad r = 1, \ldots, s$$
$$\mu_q = \frac{1}{t y_{qd}^{gu'}}, \qquad q = 1, \ldots, t$$
$$\omega_p = \frac{1}{k \bar{y}_{pd}^{bu'}} \qquad p = 1, \ldots, k$$
$$v_i = 0; \qquad i = 1, \ldots, m$$
$$u_0 = \max\{\alpha_j\} + 1$$

where:

$$\alpha_j = \sum\nolimits_{q=1}^{t} \frac{1}{s y_{qd}^{gu'}} y_{qj}^{gu'} + \sum\nolimits_{p=1}^{k} \frac{1}{s \bar{y}_{pd}^{bu'}} \bar{y}_{pj}^{bu'} \qquad j = 1, \ldots, n$$

Model (20) is also bounded. The proof is similar to that of model (19).

Theorem 4.3 The inequality $\gamma^{l*} \leq \gamma^{u*}$ holds for all DMUs.

Proof. Suppose $(\gamma_r^*, \quad \lambda_j^*)$ for $r=(1, \ldots, s)$, and $j=(1, \ldots, n)$ is an optimal solution of model (19). Hence, the inequality (22) holds.

$$\sum_{j=1}^{n} \lambda_j^* x_{ij}^{l'} \leq x_{id}^{l'} \quad i = 1, \ldots, m \tag{22}$$

We know $x_{id}^{l'} \leq x_{id}^{u'}$. Hence, inequality (23) holds.

$$\sum_{j=1}^{n} \lambda_j^* x_{ij}^{l'} \leq x_{id}^{u'} \quad i = 1, \ldots, m \tag{23}$$

Also, the inequalities (24) and (25) hold for any optimal solution of model (19).

$$\sum_{j=1}^{n} \lambda_j^* y_{qj}^{gu'} \geq \gamma_q^* y_{qd}^{gu'} \quad q = 1, \ldots, t \tag{24}$$

$$\sum_{j=1}^{n} \lambda_j^* \bar{y}_{pj}^{bu'} \geq \gamma_p^* \bar{y}_{pd}^{bu'} \quad p = 1, \ldots, k \tag{25}$$

We know that inequalities (26) and (27) always hold.

$$y_{qd}^{gu'} \geq y_{qd}^{gl'} \quad q = 1, \ldots, t \tag{26}$$

$$\overline{y}_{pd}^{bu'} \geq \overline{y}_{pd}^{bl'} \quad p=1,\ldots,k \tag{27}$$

Based on inequalities (24)-(27), inequalities (28) and (29) hold.

$$\sum_{j=1}^{n} \lambda_j^* y_{qj}^{gu'} \geq \gamma_q^* y_{qd}^{gl'} \quad q=1,\ldots,t \tag{28}$$

$$\sum_{j=1}^{n} \lambda_j^* \overline{y}_{pj}^{bu'} \geq \gamma_p^* \overline{y}_{pd}^{bl'} \quad p=1,\ldots,k \tag{29}$$

Regarding the BCC constraint of models (19) and (20), equality (30) holds for any optimal solution.

$$\sum_{j=1}^{n} \lambda_j^* = 1 \quad j=1,\ldots,n \tag{30}$$

As mentioned, we consider the constraint $\gamma_r \geq 1$ for $r=(1, 2, \ldots, s)$ in the proposed models. Hence, inequality (31) holds for any optimal solution of these models.

$$\gamma_r^* \geq 1 \quad r=1,\ldots,s \tag{31}$$

From inequalities (23) and (28)-(31), we can conclude that the optimal solution of model (19) is a feasible solution of model (20). Therefore, the inequality $\gamma^{l}* \leq \gamma^{u}*$ holds. ▪

5 Usefulness of the proposed models for QoS-aware Web service composition

Although the proposed models rank Web services based on their relative efficiency, they are not suitable for Web service composition because combining separate best services might not always satisfy a user's QoS constraints. A QoS-aware Web service composition approach is needed to orchestrate the qualified services based on the requested workflow, so as to maximize the aggregate QoS and meet user-specified global QoS constraints.

As mentioned, Web service composition is an optimization problem, and many studies have been conducted using different algorithms to solve it. In this section, we investigate the advantages of the proposed models when they are employed in the preprocessing phase of metaheuristic algorithms. We should prepare data before running these algorithms. This is carried out in two steps, as follows.

1. Weighting QoS values: As mentioned, the proposed models are integrated with interval entropy weights. Hence they select qualified Web services according to the dynamic cloud environment. Similarly, in the next phase, QoS-aware Web service composition approaches should discover the best possible compositions according to this situation. Thus we multiply the lower and upper bounds of QoS attributes by their interval entropy weights using Eq. (14). Therefore, these approaches will be able to compute

the fitness of compositions while considering the dynamic nature of the cloud environment. This is the main usefulness of the proposed models for cloud service selectiom problem.

2. Normalization of attribute values: The different measurement metrics of QoS attributes may lead to inaccurate evaluations. The QoS values should be normalized to avoid this. As mentioned, QoS attributes can be classified as desirable (positive) or undesirable (negative). Hence, the lower and upper bounds of positive attribute values are normalized to a range of 0-1 using the Eqs. (32) and (33), respectively.

$$q_{S_{ij}}^{l,k} = \begin{cases} \dfrac{q_{S_{ij}}^{l,k} - \mathrm{Min}(q_{S_{ij}}^{l,k})}{\mathrm{Max}(q_{S_{ij}}^{l,k}) - \mathrm{Min}(q_{S_{ij}}^{l,k})} & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{l,k}) - \mathrm{Min}(q_{S_{ij}}^{l,k}) \neq 0 \\ 1 & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{l,k}) - \mathrm{Min}(q_{S_{ij}}^{l,k}) = 0 \end{cases} \text{Positive attributes} \tag{32}$$

$$q_{S_{ij}}^{u,k} = \begin{cases} \dfrac{q_{S_{ij}}^{u,k} - \mathrm{Min}(q_{S_{ij}}^{u,k})}{\mathrm{Max}(q_{S_{ij}}^{u,k}) - \mathrm{Min}(q_{S_{ij}}^{u,k})} & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{u,k}) - \mathrm{Min}(q_{S_{ij}}^{u,k}) \neq 0 \\ 1 & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{u,k}) - \mathrm{Min}(q_{S_{ij}}^{u,k}) = 0 \end{cases} \text{Positive attributes} \tag{33}$$

In Eqs. (32) and (33), $q_{S_{ij}}^{l,k}$ and $q_{S_{ij}}^{u,k}$ represent the lower and upper bounds, respectively, of the kth QoS attribute of the jth Web service from the ith abstract service. Also, $Max(q_{S_{ij}}^{l,k})$ and $Min(q_{S_{ij}}^{u,k})$ are respectively the maximum $q_{S_{ij}}^{l,k}$ and minimum $q_{S_{ij}}^{u,k}$ among all qualified Web services. The lower and upper bounds of the negative attribute values are normalized to a range of 0-1 using the Eqs. (34) and (35), respectively.

$$q_{S_{ij}}^{l,k} = \begin{cases} \dfrac{\mathrm{Max}(q_{S_{ij}}^{l,k}) - q_{S_{ij}}^{l,k}}{\mathrm{Max}(q_{S_{ij}}^{l,k}) - \mathrm{Min}(q_{S_{ij}}^{l,k})} & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{l,k}) - \mathrm{Min}(q_{S_{ij}}^{l,k}) \neq 0 \\ 1 & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{l,k}) - \mathrm{Min}(q_{S_{ij}}^{l,k}) = 0 \end{cases} \text{Negative attributes} \tag{34}$$

$$q_{S_{ij}}^{u,k} = \begin{cases} \dfrac{\mathrm{Max}(q_{S_{ij}}^{u,k}) - q_{S_{ij}}^{u,k}}{\mathrm{Max}(q_{S_{ij}}^{u,k}) - \mathrm{Min}(q_{S_{ij}}^{u,k})} & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{u,k}) - \mathrm{Min}(q_{S_{ij}}^{u,k}) \neq 0 \\ 1 & \text{if} \quad \mathrm{Max}(q_{S_{ij}}^{u,k}) - \mathrm{Min}(q_{S_{ij}}^{u,k}) = 0 \end{cases} \text{Negative attributes} \tag{35}$$

Much effort has been devoted to cloud service selection using single- and multi-objective metaheuristic algorithms, such as studies based on GA [Ding, Liu, Sun et al. (2015)], NSGA-II [Chen, Dou, Li et al. (2016)], PSO [Seghir and Khababa (2018)], MOPSO [Huo, Qiu, Zhai et al. (2018)], IWO [Jatoth, Gangadharan and Fiore (2019)], MOEAD [Suciu, Pallez, Cremene et al. (2013)], Eagle [Gavvala, Jatoth, Gangadharan et al. (2019)], and ABC [Huo, Zhuang, Gu et al. (2015)]. These studies apply an integer encoding scheme to candidate compositions. representing each by an m-dimensional array $x_d = \{x_1^d, x_2^d, \ldots, x_m^d\}$. An element x_j^i of this array denotes the ith concrete service from the jth abstract service. Proposed models select attractive Web services that have the best efficiency scores. In this way, proposed models eliminate unqualified services for each abstract service and reduce the design space. Hence, employing our interval DEA models in the preprocessing phase of the metaheuristic algorithms decreases the probability of being trapped in local optima and increases the convergence rate. In Section 8, we will adopt GA as an instance of metaheuristic algorithm and confirm this. Fig. 7 illustrates how the proposed models reduce the design space by eliminating unqualified concrete services for each abstract service.

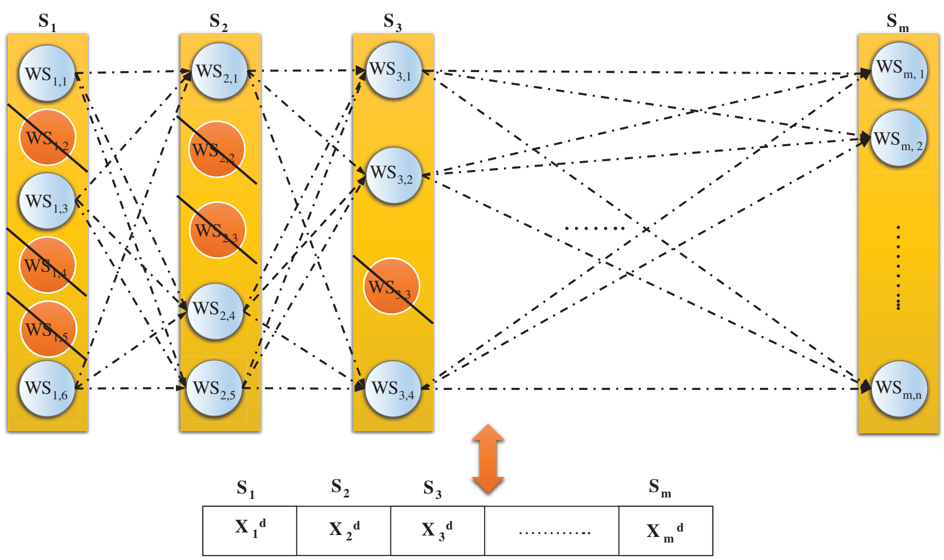


Figure 7: Eliminating unqualified concrete services

The time complexity of the DEA model is linear with the number of qualified services. Hence the time complexity of our preprocessing phase is proportional to $O(mn)$, where m is the number of abstract services, and n is the number of qualified concrete services for each abstract service. This run-time overhead can be considered as a disadvantage of the proposed solution.

Our solution can expand to a variety of applications that need effective search such as [Abdullahi, Ngadi and Abdulhamid (2016); Awad, El-Hefnawy and Abdel-kader (2015)] and Virtual Machine Placement in cloud computing [Masdari, Nabavi and Ahmadi (2016); Choudhary, Rana and Matahai (2016)].

6 Experiment

6.1 Dataset description

This study proposed a pair of models to rank Web services more precisely considering the uncertainty factor, and other factors such as user feedback techniques [Jatoth, Gangadharan and Fiore (2017)] are not included in its assumptions. Hence, we should find an appropriate dataset to conduct our experiment under the uncertainty factor.

There is currently no open dataset that includes a variety of QoS attributes to evaluate our interval DEA models. Performance of Web services is evaluated by throughput and response time attributes. Here, to be more realistic, we confine our experiment to the performance evaluation of Web services whose needed information is available in the WS-DREAM dataset#1. This dataset includes QoS values related to 5285 public Web services which monitored by 339 users. It is available for download (https://github.com/wsdream/wsdream-dataset/tree/master/dataset1), hence our experiment is reproducible.

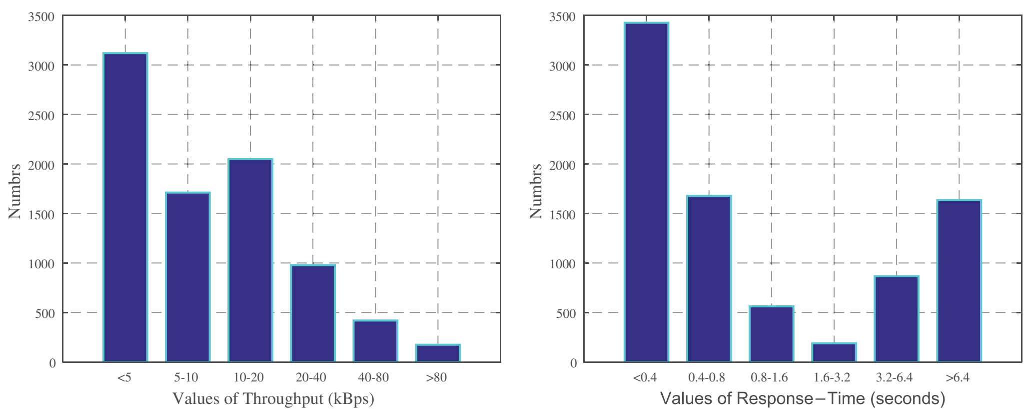
The Web services of the WS-DREAM dataset have different functionalities. We confine our selection to 25 Web services with similar search functionality. These are selected from the original dataset by keyword-search. The QoS values (throughput and response time) of these Web services can be represented as a 339×25 matrix, whose each entry is a vector that includes corresponding throughput and response time.

Tab. 1 includes a summary of the QoS values used in the experiment. Missing (out-of-range) values were removed before computing the mean values of QoS attributes for this table. These were filled with the obtained mean values to compute the standard deviation. As is seen in the table, there is a remarkable difference between the minimum and maximum values of each QoS attribute. Also, the large standard deviations values indicate considerable uncertainty in these QoS values. This confirms the necessity to find a solution to deal with the uncertainty in the Web service selection process.

Table 1: Summary of extracted dataset

Statistics	**Values**
Num. of Web service invocations	8475
Num. of service users	339
Num. of Web services	25
Num. of user countries	31
Num. of Web service countries	9
Minimum response time value	0.007 s
Maximum response time value	19.23 s
Mean of response time	3.343 s
Standard deviation of response time	4.460 s
Minimum throughput value	0.122 kbps
Maximum throughput value	428.6 kbps
Mean of throughput	13.64 kbps
Standard deviation of throughput	22.49 kbps

Value distributions of throughput and response time attributes related to the chosen 25 Web services are shown in Figs. 8(a) and 8(b), respectively, from which we observe that the major portion of response time values is less than 0.8 seconds, and the major portion of the throughput values is less than 20 kilobytes per second (kbps).



(a) (b)

Figure 8: Value distributions of the chosen Web services

We should extract the lower and upper bounds of QoS values to conduct the experiment. Tab. 2 includes the extracted values from the WS-DREAM dataset corresponding to the lower and upper bounds of the selected Web services' QoS attributes, which we use as the outputs of DMUs.

Since WS-DREAM does not contain the price of Web services, we use a synthesized dataset to conduct our experiment. In detail, we assume that the corresponding values of this missing attribute are equal together.

6.2 Efficiency evaluation

We will use the described dataset to evaluate the relative efficiency of selected Web services via the proposed models. We then compare the proposed models to the interval DEA models. Tab. 3 displays the relative importance of the QoS attributes which have been computed by the interval entropy method. In fact, this table display the last raw of the decision matrices in Figs. 4(b) and 5(a). The other rows of these matrices are shown in Tab. 2. As shown for both the lower and upper bounds, the weights of the throughput attribute are more than those of the response time attribute. We can conclude that the throughput attribute provides more information than the response time attribute for decision making. According to the equal values assigned to inputs, the entropy weights of the price attribute will be equal to 1. After computing the entropy weights, we use them to construct a new PPS by Eq. (14).

Table 2: Chosen QoS values for conducting the experiment

21.3 cm Services	**Response-time**		**Throughput**	
	y^l	y^u	y^l	y^u
1	0.409	13.32	0.600	19.56
2	0.413	13.66	0.336	19.37
3	0.377	15.91	0.502	21.22
4	0.504	13.41	0.596	15.87
5	0.398	13.58	0.589	20.10
6	0.439	19.23	0.416	18.22
7	0.488	13.75	0.581	16.39
8	0.037	7.829	0.383	81.08
9	0.083	4.230	1.891	96.39
10	0.155	6.613	0.907	38.71
11	0.095	6.612	0.604	42.11
12	0.014	5.150	0.388	142.9
13	0.007	3.353	0.596	285.7
14	0.106	7.062	0.849	56.60
15	0.093	8.428	0.711	64.52
16	0.097	15.32	0.391	61.86
17	0.031	9.502	0.210	64.52
18	0.014	6.932	0.288	142.9
19	0.012	10.47	0.191	166.7
20	0.008	2.863	0.698	250
21	0.011	2.062	0.969	181.8
22	0.007	2.098	1.429	428.5
23	0.022	6.631	0.452	136.4
24	0.349	10.27	0.152	20.06
25	0.019	6.464	0.464	157.9

Table 3: Interval entropy weights

	Response-time		**Throughput**	
	w^l	w^u	w^l	w^u
Interval entropy weights	0.022	5.618	0.066	5.755

We evaluate the relative efficiency of Web services using models (19) and (20), since the response time attribute is considered an undesirable output for Web services. These models are coded using GAMS distribution 24.1.2. In addition, we set the translation vector of undesirable outputs using Eq. (36) before running our models.

$$w_p = \max\{y_p^{bu'}\} + 1, \quad (p = 1, 2, \ldots, k) \tag{36}$$

The results are summarized in Tab. 4. The first three columns are the results of model (19), including the efficient values of response time (column γ_1^*), efficient values of throughput (column γ_2^*), and amounts of γ^{l^*} (column γ^{l^*}). The next three columns are the corresponding results of model (20).

As is seen, all of the obtained efficiency scores are greater than or equal to 1 since we evaluate Web services using the output-oriented models. A Web service is efficient from the optimistic viewpoint, if values of both γ_1^* and γ_2^* are equal to 1 in model (19). In Tab. 4, only Web service 22 is efficient from the optimistic viewpoint. The other Web services are inefficient since their efficiency scores are greater than 1. Web service 4 has the worst efficiency among the results of model (19) because its values of γ^{l^*} is more than those of the other optimistic efficiency scores. In the same way, Web services 9 and 24 respectively have the best and worst efficiency scores among the results of model (20). We observe that the computed results confirm theorems in Section 4.2, since all solutions are feasible and bounded, and we always have $\gamma^{l}* \leq \gamma^{u}*$.

As mentioned in Section 4.2, an inefficient Web service can be converted into an efficient one through the proportional increase on its outputs while the proportion of its inputs remains fixed. Here, we compute the output insufficiency/redundancy rates of inefficient Web services. As mentioned in Section 4.2, the insufficiency/redundancy rate of the rth output is γ_r^*–1 in model (19), where r=(1, 2, …, s). For example, consider Web service 8 in Tab. 4. The output redundancy rate of its response time is 0.00149 s, and the output insufficiency rate of its throughput is 4.28614 kbps. It means this Web service will become efficient if these improvements are applied to its QoS attributes. As another example, consider the Web service 13 in Tab. 4. There is no need to make any improvements in its response time attribute since its redundancy rate is 0.

Next, we evaluate the selected Web service set using the interval DEA models. For this purpose, we run the modified versions of models (2) and (3) considering undesirable outputs.

The results of the lower and upper bounds of the interval DEA models are incomplete. Hence, we integrate their results using the geometric average. The results are shown in Tab. 5. The first two columns are the results of proposed models that include the overall efficiency scores of the Web services (column Geom) and their ranking results (column Rank). Similarly, the next two columns include results of the interval DEA models. Fig. 9 shows the Spearman's rank correlation coefficient between the ranking of the

Table 4: Evaluation results of proposed models

Services	Lower bound			Upper bound		
	γ_1^*	γ_2^*	γ'^*	γ_1^*	γ_2^*	γ''^*
1	1.02028	21.9121	11.466	740.512	62560.6	31651
2	1.02049	22.1270	11.574	778.834	102558	51668
3	1.01864	20.1979	10.608	1184.48	74773.6	37979
4	1.02520	27.0069	14.016	750.285	62980.4	31865
5	1.01972	21.3234	11.172	769.465	63728.9	32249
6	1.02183	23.5236	12.273	5116.94	90231.6	47674
7	1.02436	26.1501	13.587	789.651	64606.4	32698
8	1.00149	5.28614	3.1438	412.623	98006.1	49209
9	1.00377	4.44652	2.7251	319.809	19850.0	10085
10	1.00737	11.0721	6.0397	375.776	41385.2	20880
11	1.00437	10.1781	5.5912	375.748	62146.3	31261
12	1.00035	2.99930	1.9998	339.320	96743.1	48541
13	1	1.50018	1.2501	303.190	62980.4	31642
14	1.00492	7.57244	4.2887	388.589	44212.4	22300
15	1.00427	6.64290	3.8236	433.566	52793.7	26614
16	1.00447	6.92855	3.9665	1042.15	96000.9	48522
17	1.00119	6.64290	3.8220	476.971	178744	89610
18	1.00035	2.99930	1.9998	384.790	130334	65359
19	1.00025	2.57109	1.7857	524.277	196525	98525
20	1.00005	1.71440	1.3572	294.636	53777.0	27036
21	1.00020	2.35754	1.6789	281.646	38737.2	19509
22	1	1	1	282.205	26267.6	13275
23	1.00074	3.14223	2.0715	376.273	83045.0	41711
24	1.01720	21.3659	11.192	513.749	246950	123732
25	1.00059	2.71438	1.8575	371.709	80897.3	40635

interval DEA models and the proposed models, which is 0.897 (slope of trendline). The p-value is 1.2835E-9. We can conclude that this correlation is significant.

Finally, we employ Sengupta's approach instead of the geometric average in the above experiment. In this experiment, the Spearman's rho and p-value are modified to 0.818

Table 5: Comparison of results

Services	Our Models		Interval DEA	
	Geom	**Rank**	**Geom**	**Rank**
1	0.001660	19	0.578701	18
2	0.001293	24	0.564225	21
3	0.001575	20	0.457952	24
4	0.001496	22	0.573547	19
5	0.001666	18	0.567869	20
6	0.001307	23	0.219986	25
7	0.001500	21	0.559284	22
8	0.002542	13	0.782482	13
9	0.006032	2	0.887810	5
10	0.002816	11	0.817563	10
11	0.002392	14	0.818811	9
12	0.003210	9	0.863417	6
13	0.005028	5	0.913518	4
14	0.003234	8	0.804954	12
15	0.003135	10	0.762301	14
16	0.002279	16	0.491637	23
17	0.001709	17	0.727901	15
18	0.002766	12	0.810772	11
19	0.002384	15	0.694643	17
20	0.005220	4	0.926721	3
21	0.005525	3	0.947741	1
22	0.008679	1	0.946900	2
23	0.003402	7	0.819743	8
24	0.000850	25	0.695835	16
25	0.003640	6	0.824791	7

and 6.0153E-7, respectively. The obtained results again represent that there is a significant correlation between their rankings, and Fig. 10 confirms this.

By comparing the results of these empirical studies, it can be concluded that there is a close relationship between the proposed models and the interval DEA models, although, as seen in Tab. 5, there are considerable differences in their rankings which leads to different service

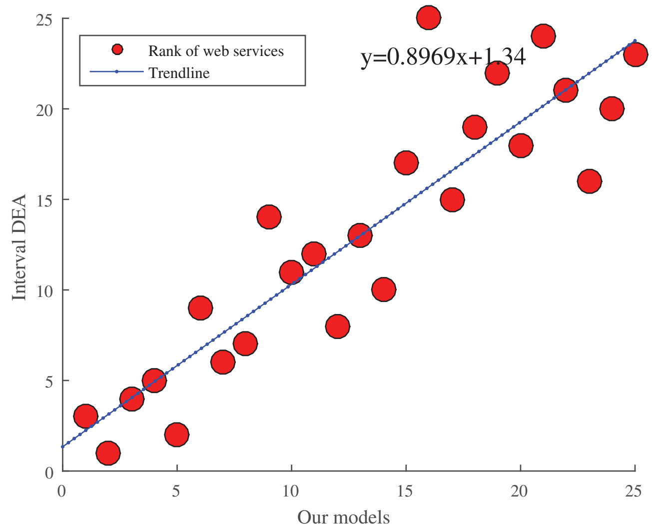


Figure 9: Obtained Spearman's rank correlation coefficient using the geometric average

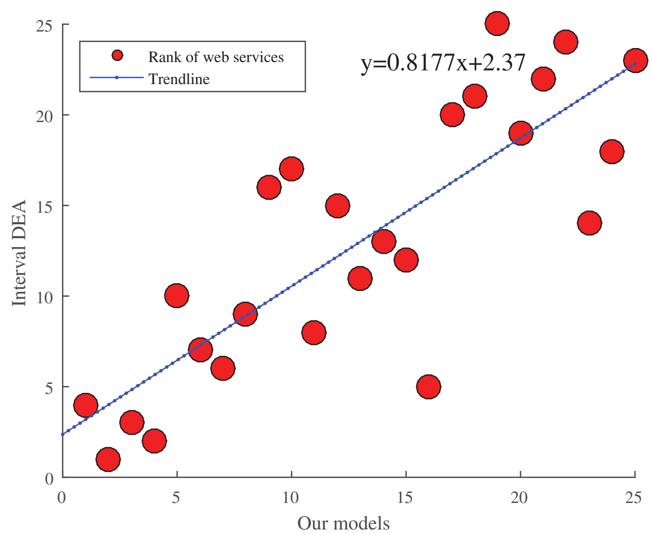


Figure 10: Obtained Spearman's rank correlation coefficient using the Sengupta's approach

recommendations. For example, the proposed models select Web service 22 and the interval DEA models select Web service 21.

7 Analysis of proposed models

Here, we aim to analyze the sensitivity of the computed results to data variations, and we present a comparative analysis of the proposed models and the interval DEA models. In this comparison, we investigate the motivation of presenting our interval DEA models based on Russell's model to rank Web services.

7.1 Sensitivity analysis of proposed models to weight variations

Sensitivity analysis determines the robustness or stability of the proposed models. We determine their robustness when the interval entropy weights are varied. For this purpose, we change the interval entropy weights while fixing all of the QoS values at their current levels. Then we find the impact of these variations in the Web service ranking. We conducted two experiments.

First, we gradually changed the difference between the lower and upper bounds of response time values for all of the Web services in Tab. 2 to obtain several interval entropy weights. Throughput values were assumed to be fixed at their current levels. Tab. 6 shows the obtained weights. We evaluate the Web services in Tab. 2, using models (19) and (20) for these weights. Fig. 11 shows the results, where the *x*- and *y*-axes represent Web services and their efficiency scores, respectively. Green bars correspond to the interval entropy weights of Tab. 3, and the other three colors correspond to the interval entropy weights included in Tab. 6. As is seen, the best Web service Web service (#22) does not change despite the different weights. The ranking of other Web services follows this trend as

Table 6: Interval entropy weights for response time variation

Interval entropy Weights	**Response-time**		**Throughput**	
	w^l	w^u	w^l	w^u
1	0.041	0.472	0.120	5.755
2	0.030	2.563	0.090	5.755
3	0.022	5.831	0.065	5.755

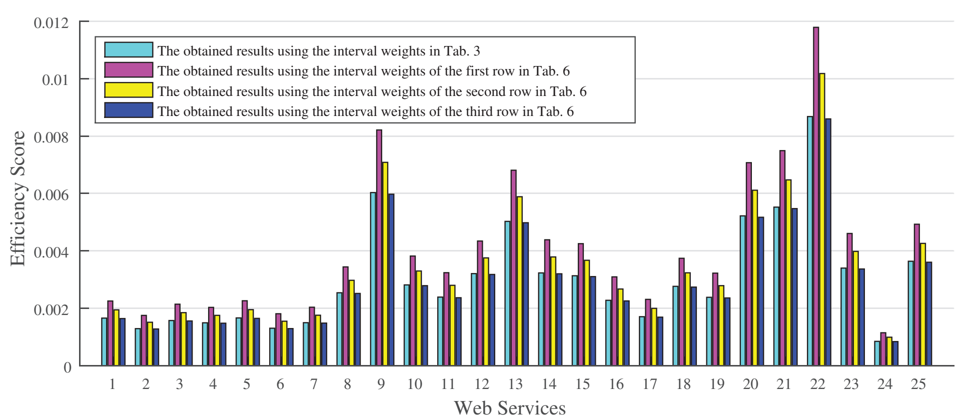


Figure 11: Sensitivity analysis of proposed models to response time variations

well. We can conclude that no significant change will be made in the ranking of Web services if different weights are used.

In the second experiment, we gradually changed the difference between the lower and upper bounds of throughput values for all of the Web services in Tab. 2 to obtain several interval entropy weights. Response time were assumed to be fixed at their current levels. The other steps of this experiment were similar to those of the first experiment. The obtained weights are presented in Tab. 7. Fig. 12 shows the results, which are similar to those in Fig. 11. As is seen, the best Web service (#22) does not change despite the different weights. The ranking of other Web services follows this trend as well. From these experiments, we can conclude that our models are robust to interval entropy weight variations.

Table 7: Interval entropy weights for throughput variations

Interval entropy weights	Response-time		Throughput	
	w^l	w^u	w^l	w^u
1	0.039	5.618	0.116	0.834
2	0.032	5.618	0.096	2.173
3	0.025	5.618	0.075	4.372

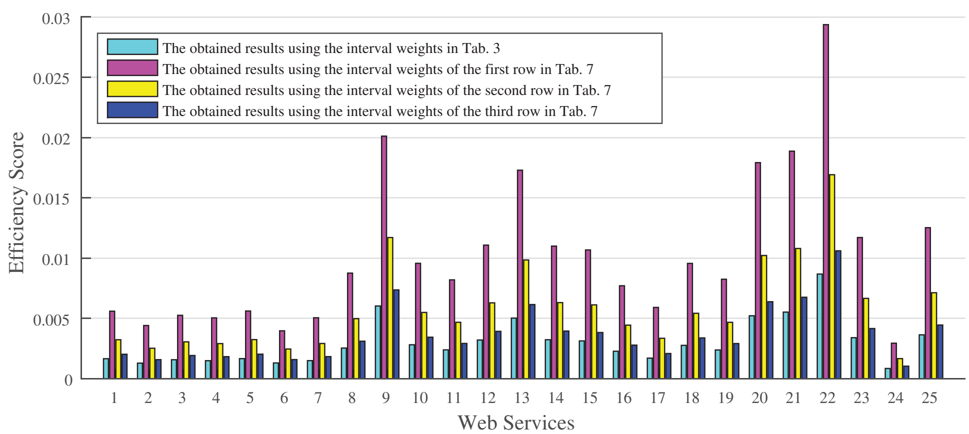


Figure 12: Sensitivity analysis of proposed models to throughput variations

7.2 Sensitivity analysis of proposed models to changes in available Web services

Given the large scale and dynamic environment of the internet, candidate services may still experience numerous sudden failures and become unavailable, and new candidate services may be launched after the recovery process [Wang, Zheng, Sun et al. (2011); Sharma, Javadi, Si et al. (2016)].

Here, we conduct another sensitivity analysis, to test the robustness of our models after recovering the faulty services. From the WS-DREAM dataset, we used a keyword search to find another 10 Web services that perform search functionality. The first four columns of Tab. 8 include the lower and upper bounds of their QoS values. Such as in Section 6.2, we assume equal values for price items. In each experiment, we add a Web service from Tab. 8 to the Web services of Tab. 2 and rank them using models (19) and (20). After each experiment, we eliminate the added Web service from the results and compare them with the results of Tab. 5. The last column of Tab. 8 includes the Spearman's rank correlation coefficients of these comparisons. We observe that the order of results remains the same (Spearman's rho=1) in most of experiments, and no significant change In this section, we discussed that the robustness of the proposed models against failure of candidate services is an essential feature, which should be considered in the ranking of Web services. Then, we demonstrated that the proposed models are almost robust against these failures, which is an important advantage of the proposed models.

Table 8: Spearman's rank correlation coefficient

Services	**Response-time**		**Throughput**		**Spearman's rho**
	y^l	y^u	y^l	y^u	
1	0.414	13.87	0.576	19.32	1
2	0.157	18.93	0.317	38.22	1
3	0.417	18.57	0.699	31.18	1
4	0.016	3.078	18.52	1000	0.984
5	0.045	14.19	4.018	826.1	0.982
6	0.010	10.63	0.188	200	1
7	0.012	10.47	0.191	166.7	1
8	0.036	2.860	2.097	166.7	1
9	0.037	7.829	0.383	81.08	1
10	0.005	9.928	0.201	400	1

7.3 Comparative analysis of proposed models and interval DEA models

As mentioned, we constructed our interval DEA models based on Russell's model to enhance their accuracy against QoS variations. To study the effect of this issue in the Web service ranking, we present a comparative analysis of the interval DEA models and proposed models. We evaluate a particular Web service using these models while varing its QoS values, and we analyze the impact of these variations on its relative efficiency score. We assume that data for the other Web services are fixed at their current values, and the weights of QoS attributes are the same. We selected Web service 21 of Tab. 2 for the first experiment. We changed the lower and upper bounds of its response time values from 2 to 10 s, with a step value of 2. We assume that its throughput values are fixed at their current levels. Fig. 13 shows the results. The blue and orange lines represent the ranking of this Web service as obtained by the interval DEA models and proposed models, respectively. We observe that the ranking of this Web service drops from the 4th position to the 22nd position, while its ranking decreased less using the proposed models (see Fig. 13 for details).

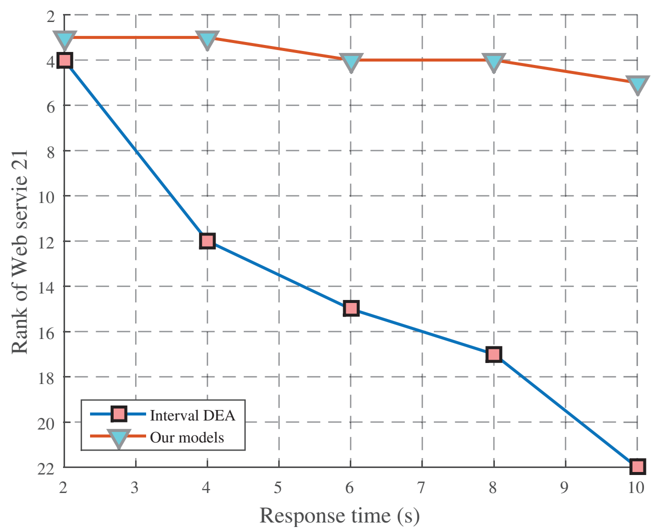


Figure 13: Effect of response time variation in ranking of Web service 21

Tab. 9 includes the evaluation results of this Web service using the interval DEA models and proposed models. The first column (column Response time) includes the values assigned to the lower and upper bounds of its response time attribute. The other columns include the results of the interval DEA models (columns $\varphi^{l}*$ and $\varphi^{u}*$) and the results of the models (19) and (20) (columns γ_1^*, γ_2^*, $\gamma^{l}*$ and $\gamma^{u}*$). As is seen, the efficient value of response time (γ_1^*) is always less than that of throughput (γ_2^*) when response time values are varied from 2 to 10. As we know, the interval DEA models increase/decrease the outputs/inputs at the same rate. Therefore, these models only use γ_1^* to compute the efficiency score of Web service 21 in the mentioned range. For example, we have $\varphi^{l}*=\gamma_1^*=1.109$ and $\varphi^{u}*=\gamma_1^*=1.109$ when the response time is 2 s. Hence we lose some useful information related to throughput in the ranking process, and we observe a considerable reduction in the rank of Web service 21, although this is one of the best among the Web services in Tab. 2 as regards throughput. The proposed models dedicate separate efficient values to each QoS attribute and compute $\gamma^{l}*$ and $\gamma^{u}*$ as their arithmetic average. For example, as seen in Tab. 9, we have $\gamma^{l}*=\left(\frac{1.109+2.358}{2}\right)=1.733$ and $\gamma^{u}*=\left(\frac{1.109+442.3}{2}\right)=221.7$ when the

Table 9: Evaluation results of Web service 21 when response time is varied from 2 to 10 s

Response time	Lower bound				Upper bound			
	φ^{l*}	γ_1^*	γ_2^*	γ^{l*}	φ^{u*}	γ_1^*	γ_2^*	γ^{u*}
2	1.109	1.109	2.358	1.733	1.109	1.109	442.3	221.7
4	1.246	1.246	2.358	1.802	1.246	1.246	442.3	221.8
6	1.421	1.421	2.358	1.889	1.421	1.421	442.3	221.9
8	1.654	1.654	2.358	2.006	1.654	1.654	442.3	222.0
10	1.977	1.977	2.358	2.167	1.977	1.977	442.3	222.1

response time is 2 s. We can conclude that the proposed models can use all of the information in the ranking process, and we observe a lesser reduction in its ranking.

For the second experiment, we selected Web service 24 from Tab. 2, and we simultaneously changed the lower and upper bounds of its throughput from 1 to 15 kbps. We assume that its response time values are fixed at their current levels. Fig. 14 shows the results. Tab. 10 includes the evaluation results of Web service 24 using the interval DEA models and the proposed models. The first column (Throughput) includes the values assigned to the lower and upper bounds of its throughput attribute. The specifications of the remaining columns are analogous to those in Tab. 9 As seen in Tab. 10, γ_1^* is always less than γ_2^* when throughput values vary from 1 to 15 kbps. As before, the interval DEA models only use γ_1^* to compute the efficiency score of Web service 24 in the mentioned range. For example, as seen in Tab. 10, we have φ^{l*}=γ_1^*=1.017 and φ^{u*}=γ_1^*=2.030 when

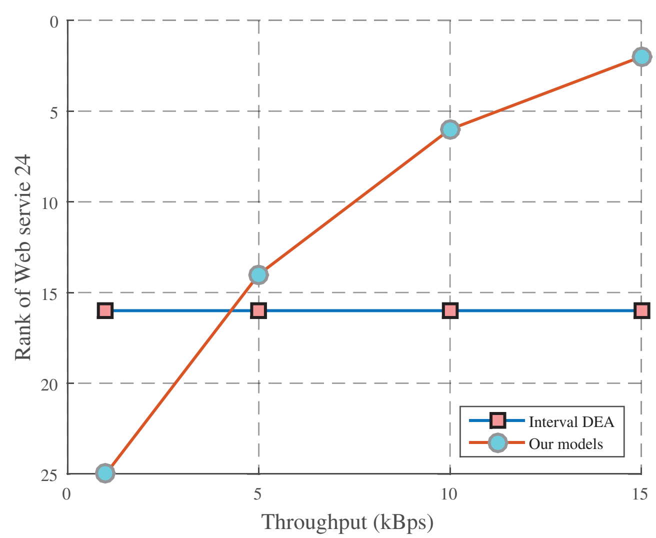


Figure 14: Effect of throughput variation in ranking of Web service 24

Table 10: Evaluation results of Web service 24 when throughput is varied from 1 to 15 kbps

Throughput	Lower bound				Upper bound			
	φ^{l*}	γ_1^*	γ_2^*	γ^{l*}	φ^{u*}	γ_1^*	γ_2^*	γ^{u*}
1	1.017	1.017	428.6	214.8	2.030	2.030	428.6	215.3
5	1.017	1.017	85.72	43.37	2.030	2.030	85.72	43.88
10	1.017	1.017	42.86	21.94	2.030	2.030	42.86	22.45
15	1.017	1.017	28.57	14.80	2.030	2.030	28.57	15.30

throughput is equal to 1 kbps. Therefore, despite increasing throughput values from 1 to 15 kbps, we observe that the efficiency score and ranking of Web service 24 do not change. The proposed models remove this drawback by dedicating separate efficient values to each QoS attribute. Thus our interval DEA models can utilize all of the efficient values to compute $\gamma^{l}*$ and $\gamma^{u}*$. For example, as seen in Tab. 9, we have $\gamma^{l}*= \left(\frac{1.017+428.6}{2}\right)$=214.8 and $\gamma^{u}*= \left(\frac{2.030+428.6}{2}\right)$=215.3 when throughput is equal to 1 kbps.

To summarize, we discussed the another advantage of the proposed models in this section. In detail, we illustrated that the proposed models are stable, while the interval DEA models are unstable, and they may show irrational changes in their results against QoS variations. For example, Fig. 13 shows that the interval DEA models may demonstrate significant changes in their results against response time variations but, as shown in Fig. 14, they are quite indifferent against throughput changes. Hence, given the uncertainty of the internet environment, it is recommended to evaluate Web services using the proposed models, which change the efficiency of Web services proportionately to the QoS variations.

8 Usefulness of proposed models for GA

This study provided a solution to discover the best possible compositions according to the dynamic nature of the cloud environment. For this purpose, we employed our models in the preprocessing phase of the metaheuristic algorithms to reduce the invalid design space. This enhances the efficiency of the resultant compositions by decreasing the probability of being trapped in local optima.

In this section, we choose canonical GA [Canfora, Di Penta, Esposito et al. (2005)] as an example of a metaheuristic algorithms to evaluate the effectiveness of our solution. We conducted experiments to compare our solution with canonical GA. For each experiment, we set the population size to 10. We use uniform crossover method [Lin, Sir and Pasupathy (2013)] to ensure a wide exploration of the design space. This method exchanges information between two parent solutions randomly to produce two offspring. In other words, each genome of offspring is selected from one of the corresponding genomes of the parent solution with a 50% probability. The crossover probability is 0.8 and the rate of mutation is 0.1.

The experiments were coded in MATLAB R2015a and performed on a laptop computer with an Intel Core i7 processor at 2.5 GHz, with 8 GB RAM under Windows 10.

Fig. 15 provides an example of online shopping to demonstrate how the proposed models help GA to enhance the fitness of the resultant compositions. To conduct our experiment, we selected 180 Web services from WS-DREAM by a keyword-based method for the authentication and search functionalities. We only identified 22 Web services with finance functionality in this dataset. We generated 180 QoS values using controlled random generation methodology in the range of finance Web services. We evaluated the generated data using the Shapiro-Wilk test, which revealed that these QoS values were distributed normally. We multiplied the lower and upper bounds of QoS values by their interval entropy weights to find the best resultant compositions according to the uncertainty of the cloud environment. As mentioned, the WS-DREAM dataset does not contain the price attribute. Hence, we assume its values are 0.

Figure 15: Online shopping workflow

Tab. 11 shows the manner of measuring the lower and upper bounds of the aggregate QoS values. Only the sequential composite model is considered corresponding to the online shopping workflow pattern. Note that, l and u indicate the lower and upper bounds of the QoS values, and n is the number of Web services.

The fitness function for an individual t is defined as the arithmetic average of Eqs. (37) and (38). In other words, we assign equal weights to the lower and upper bounds of the fitness function.

$$F^l(t) = \frac{\text{Throughput}^l(t)}{\text{Price}^l(t) + \text{Responsetime}^l(t)} \tag{37}$$

$$F^u(t) = \frac{\text{Throughput}^u(t)}{\text{Price}^u(t) + \text{Responsetime}^u(t)} \tag{38}$$

We evaluated the performance of our solution by varying the number of qualified and abstract services. As shown in Fig. 16, we increased the path length of the online shopping workflow to cases with 25, 50, 75, and 100 abstract services by invoking the "search" Web service. Then we conducted a set of experiments for each case by varying the number of qualified services for each abstract service from 10 to 170 with a step value of 20.

In each experiment, the initial generation was randomly populated from the qualified Web services. Then GA was executed 50 times and the average fitness values of the 10 best compositions were computed. Fig. 17 depicts the final results. The last column in each figure shows the average fitness values of the 10 best resultant compositions identified by

Table 11: QoS aggregation functions for sequential composition model

Statistics	Values
Price^l	$\sum_{j=1}^{n} q_{S_i}^{l,1}$
Response time^l	$\sum_{j=1}^{n} q_{S_i}^{l,2}$
Throughput^l	$\min_{j=1}^{n}(q_{S_i}^{l,3})$
Price^u	$\sum_{j=1}^{n} q_{S_i}^{u,1}$
Response time^u	$\sum_{j=1}^{n} q_{S_i}^{u,2}$
Throughput^u	$\min_{j=1}^{n}(q_{S_i}^{u,3})$

$q_{S_i}^{l,t}$, $q_{S_i}^{u,t}$, $1 \le t \le 3$ show price, response time and throughput of concrete service S_i.

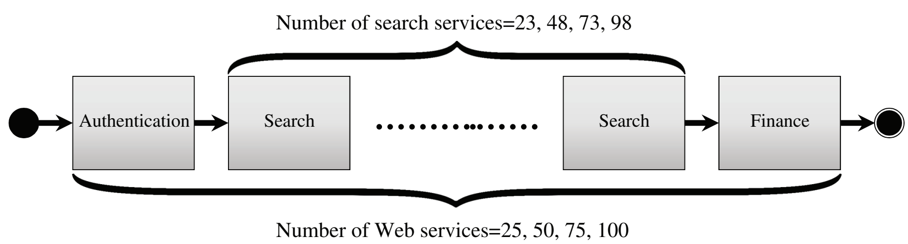


Figure 16: Varying the path length of online shopping workflow

canonical GA. The other columns represent the results of our solution when the number of qualified services changes from 10 to 170. As is seen, our solution is better than canonical GA based on the average fitness values since GA searches the larger design space.

As we know, the efficiency of resultant compositions depend on the size of the design space, and also the size of the design space changes based on the workflow length. Hence, we want to forecast how the change in the workflow length will affect the effectiveness of our solution. For this purpose, we use linear regression method [Montgomery, Peck and Vining (2015)] to determine the correlation between fitness of the resultant compositions and the workflow length. In detail, we need to find the equation of the best-fit trendline for the obtained results. The slope of this regression line represents the rate of change in efficiency score as the number of qualified services changes. The greater the magnitude of the slope, the steeper the line and the greater the rate of change. We can see that in Fig. 17, this slope decreases with increasing the workflow length. This issue indicates that the effectiveness of our solution has been reduced by increasing the workflow length. For example, the average fitness values is improved by 0.2 scale, when the

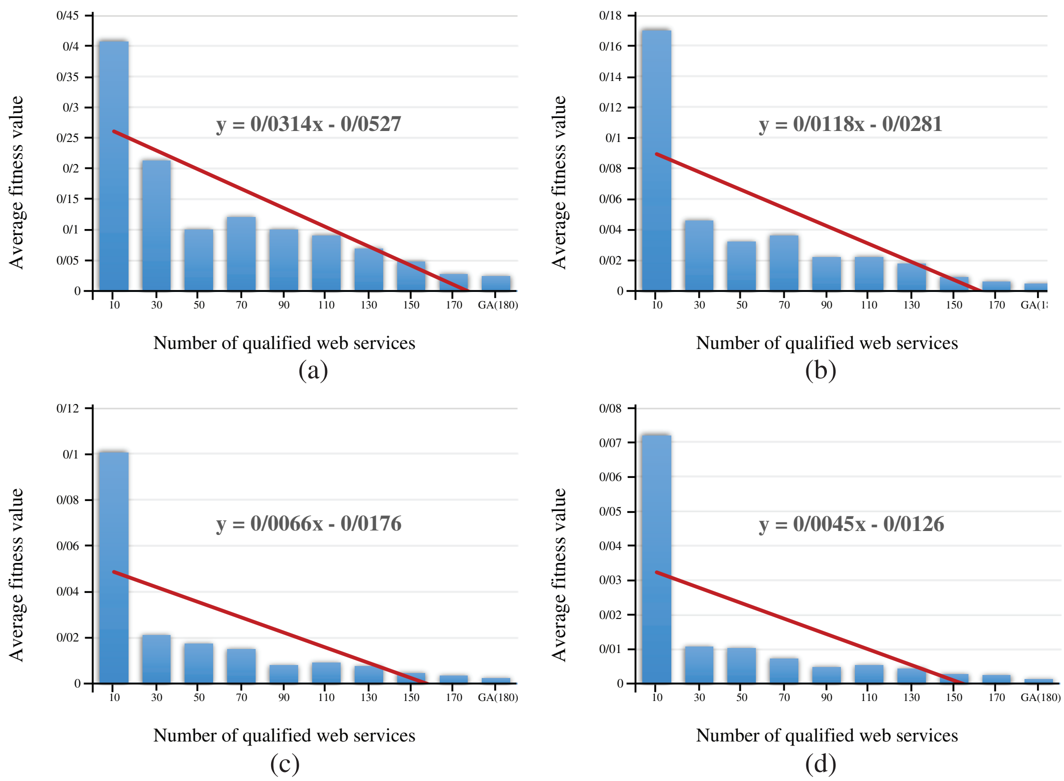


Figure 17: The average fitness of the 10 best compositions resultant from our solution and canonical GA

number of qualified services decreased from 30 to 10 in Fig. 17(a), but this improvement is about only 0.06 scale in Fig. 17(d).

Also, if we ignore the scale of Fig. 17, we observe that our solution is more effective when the path length of the workflow is longer. For example, the average fitness values improved almost seven times when the number of qualified services decreased from 30 to 10 in Fig. 17 (d), but this improvement was only about twice in Fig. 17(a).

In this section, we demonstrated that the proposed models enhance the efficiency of the resultant compositions when they are used in the preprocessing phase of the canonical GA. This issue can be considered as another advantage of the proposed models.

Moreover, we can define the user global QoS constraints in this experiment by considering the penalty-factor [Ding, Liu, Sun et al. (2015)] for the fitness function or dropping unsuitable offspring from the population [Que, Zhong, Chen et al. (2018)]. Also, we can classify the user global QoS constraints using data mining techniques to find the appropriate number of qualified services for each class.

9 Conclusion

The rapid proliferation of Web services has led to the dilemma of selecting the best Web service from many functionally equivalent candidates. In this paper, we assumed continuous interval values for QoS attributes because of the dynamic cloud environment. Based on this, we proposed modified interval DEA models to select the most suitable Web service considering the uncertainty of the internet in the presence of desirable and undesirable outputs. We constructed our models by integration of the interval DEA models and the interval entropy weights. We modified the constructed models based on the BCC and simple Russell's model.

After comparing the proposed models and the interval DEA models, we concluded that the correlation between their rankings is significant but they recommend different Web services. In addition, we found that unlike the proposed models, the interval DEA models lose some useful information in the ranking process. We performed sensitivity analyses for the interval entropy weight variations and changes in the available Web services, from which we observed that our models are almost robust. As a result, we recommend selecting the best Web service using the proposed models, which are customized according to the dynamic internet environment. Also, we presented a solution to obtain the best possible compositions based on the uncertainty of the internet by employing the proposed models in the preprocessing phase of the metaheuristic algorithms. We demonstrated that the proposed models enhance the fitness of the resultant compositions when employed in the preprocessing phase of GA as an instance of metaheuristic algorithms. In our future work, we plan to work on neural network models to eliminate unqualified concrete services for the QoS-aware Web service composition problem.

References

Abdullahi, M.; Ngadi, M. A.; Abdulhamid, S'i. M. (2016): Symbiotic organism search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems*, vol. 56, pp. 640-650. DOI 10.1016/j.future.2015.08.006.

Al-Faifi, A.; Song, B.; Hassan, M. M.; Alamri, A.; Gumaei, A. (2019): A hybrid multi criteria decision method for cloud service selection from smart data. *Future Generation Computer Systems*, vol. 93, pp. 43-57. DOI 10.1016/j.future.2018.10.023.

Anvari, A.; Zulkifli, N.; Sorooshian, S.; Boyerhassani, O. (2014): An integrated design methodology based on the use of group AHP-DEA approach for measuring lean tools efficiency with undesirable output. *International Journal of Advanced Manufacturing Technology*, vol. 70, no. 9-12, pp. 2169-2186. DOI 10.1007/s00170-013-5369-z.

Awad, A.; El-Hefnawy, N.; Abdel-kader, H. (2015): Enhanced particle swarm optimization for task scheduling in cloud computing environments. *Procedia Computer Science*, vol. 65, pp. 920-929. DOI 10.1016/j.procs.2015.09.064.

Azadi, M.; Emrouznejad, A.; Ramezani, F.; Hussain, F. K. (2019): Efficiency measurement of cloud service providers using network data envelopment analysis. *IEEE Transactions on Cloud Computing*. DOI 10.1109/TCC.2019.2927340.

Banker, R. D.; Charnes, A.; Cooper, W. W. (1984): Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science*, vol. 30, no. 9, pp. 1078-1092. DOI 10.1287/mnsc.30.9.1078.

Canfora, G.; Di Penta, M.; Esposito, R.; Villani, M. L. (2005): An approach for QoS-aware service composition based on genetic algorithms. *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, *Washington, DC, USA*, pp. 1069-1075.

Chakraverty, S. (2014): *Mathematics of Uncertainty Modeling in the Analysis of Engineering and Science Problems*. IGI Global, Hershey, PA.

Chanas, S.; Zielinski, P. (1999): Ranking fuzzy interval numbers in the setting of random sets-further results. *Information Sciences*, vol. 117, no. 3-4, pp. 191-200. DOI 10.1016/S0020-0255(99)00027-4.

Charnes, A.; Cooper, W. W.; Rhodes, E. (1978): Measuring the efficiency of decision making units. *European Journal of Operational Research*, vol. 2, no. 6, pp. 429-444. DOI 10.1016/0377-2217(78)90138-8.

Chen, F.; Dou, R.; Li, M.; Wu, H. (2016): A flexible QoS-aware web service composition method by multi-objective optimization in cloud manufacturing. *Computers & Industrial Engineering*, vol. 99, pp. 423-431. DOI 10.1016/j.cie.2015.12.018.

Choudhary, A.; Rana, S.; Matahai, K. (2016): A critical analysis of energy efficient virtual machine placement techniques and its optimization in a cloud computing environment. *Procedia Computer Science*, vol. 78, no. C, pp. 132-138. DOI 10.1016/j.procs.2016.02.022.

Cooper, W. W.; Seiford, L. M.; Tone, K. (2007): *Data Envelopment Analysis: a Comprehensive Text with Models, Applications, References and DEA-solver Software*. Springer, New York.

Ding, Z.; Liu, J.; Sun, Y.; Jiang, C.; Zhou, M. (2015): A transaction and QoS-aware service selection approach based on genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1035-1046. DOI 10.1109/TSMC.2015.2396001.

Eraslan, E.; Ic, Y. T. (2019): An improved decision support system for ABC inventory classification. *Evolving Systems*, vol. 1, pp. 1-14.

Garg, S. K.; Versteeg, S.; Buyya, R. (2013): A framework for ranking of cloud computing services. *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012-1023. DOI 10.1016/j.future.2012.06.006.

Gavvala, S. K.; Jatoth, C.; Gangadharan, G.; Buyya, R. (2019): QoS-aware cloud service composition using eagle strategy. *Future Generation Computer Systems*, vol. 90, pp. 273-290. DOI 10.1016/j.future.2018.07.062.

Gireesha, O.; Somu, N.; Krithivasan, K.; Sriram, V. S. S. (2020): IIVIFS-WASPAS: an integrated multi-criteria decision-making perspective for cloud service provider selection.

Future Generation Computer Systems, vol. 103, pp. 91-110. DOI 10.1016/j.future.2019.09.053.

Gireesha, O.; Somu, N.; Raman, M. G.; Reddy, M. S.; Kirthivasan, K. et al. (2020): WNN-EDAS: a wavelet neural network based multi-criteria decision-making approach for cloud service selection. *Computational Intelligence in Pattern Recognition*. pp. 853-865. Springer, New York.

Goscinski, A.; Brock, M. (2010): Toward dynamic and attribute based publication, discovery and selection for cloud computing. *Future Generation Computer Systems*, vol. 26, no. 7, pp. 947-970. DOI 10.1016/j.future.2010.03.009.

Hasnain, M.; Thaheem, M. J.; Ullah, F. (2018): Best value contractor selection in road construction projects: ANP-based decision support system. *International Journal of Civil Engineering*, vol. 16, no. 6, pp. 695-714. DOI 10.1007/s40999-017-0199-2.

He, F.; Zhang, Q.; Lei, J.; Fu, W.; Xu, X. (2013): Energy efficiency and productivity change of China's iron and steel industry: accounting for undesirable outputs. *Energy Policy*, vol. 54, pp. 204-213. DOI 10.1016/j.enpol.2012.11.020.

Huang, A. F.; Lan, C. W.; Yang, S. J. (2009): An optimal QoS-based web service selection scheme. *Information Sciences*, vol. 179, no. 19, pp. 3309-3322. DOI 10.1016/j.ins.2009.05.018.

Huo, Y.; Qiu, P.; Zhai, J.; Fan, D.; Peng, H. (2018): Multi-objective service composition model based on cost-effective optimization. *Applied Intelligence*, vol. 48, no. 3, pp. 651-669. DOI 10.1007/s10489-017-0996-y.

Huo, Y.; Zhuang, Y.; Gu, J.; Ni, S.; Xue, Y. (2015): Discrete gbest-guided artificial bee colony algorithm for cloud service composition. *Applied Intelligence*, vol. 42, no. 4, pp. 661-678. DOI 10.1007/s10489-014-0617-y.

Hussain, A.; Chun, J.; Khan, M. (2020): A novel framework towards viable cloud service selection as a service (cssaas) under a fuzzy environment. *Future Generation Computer Systems*, vol. 104, pp. 74-91. DOI 10.1016/j.future.2019.09.043.

Jahanshahloo, G. R.; Lotfi, F. H.; Davoodi, A. (2009): Extension of topsis for decision-making problems with interval data: interval efficiency. *Mathematical and Computer Modelling*, vol. 49, no. 5-6, pp. 1137-1142. DOI 10.1016/j.mcm.2008.07.009.

Jahed, R.; Amirteimoori, A.; Azizi, H. (2015): Performance measurement of decision-making units under uncertainty conditions: an approach based on double frontier analysis. *Measurement*, vol. 69, pp. 264-279. DOI 10.1016/j.measurement.2015.03.014.

Jatoth, C.; Gangadharan, G.; Buyya, R. (2019): Optimal fitness aware cloud service composition using an adaptive genotypes evolution based genetic algorithm. *Future Generation Computer Systems*, vol. 94, pp. 185-198. DOI 10.1016/j.future.2018.11.022.

Jatoth, C.; Gangadharan, G.; Fiore, U. (2017): Evaluating the efficiency of cloud services using modified data envelopment analysis and modified super-efficiency data envelopment analysis. *Soft Computing*, vol. 21, no. 23, pp. 7221-7234. DOI 10.1007/s00500-016-2267-y.

Jatoth, C.; Gangadharan, G.; Fiore, U. (2019): Optimal fitness aware cloud service composition using modified invasive weed optimization. *Swarm and Evolutionary Computation*, vol. 44, pp. 1073-1091. DOI 10.1016/j.swevo.2018.11.001.

Jatoth, C.; Gangadharan, G.; Fiore, U.; Buyya, R. (2019): Selcloud: a hybrid multi-criteria decision-making model for selection of cloud services. *Soft Computing*, vol. 23, no. 13, pp. 4701-4715. DOI 10.1007/s00500-018-3120-2.

Jula, A.; Sundararajan, E.; Othman, Z. (2014): Cloud computing service composition: a systematic literature review. *Expert Systems with Applications*, vol. 41, no. 8, pp. 3809-3824. DOI 10.1016/j.eswa.2013.12.017.

Karimi, M. B.; Isazadeh, A.; Rahmani, A. M. (2017): QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm. *Journal of Supercomputing*, vol. 73, no. 4, pp. 1387-1415. DOI 10.1007/s11227-016-1814-8.

Karsak, E. E. (2008): Using data envelopment analysis for evaluating flexible manufacturing systems in the presence of imprecise data. *International Journal of Advanced Manufacturing Technology*, vol. 35, no. 9-10, pp. 867-874. DOI 10.1007/s00170-006-0765-2.

Kumar, R. R.; Kumar, C. (2017). Designing an efficient methodology based on entropy-topsis for evaluating efficiency of cloud services. *Proceedings of the 7th International Conference on Computer and Communication Technology*, pp. 117-122.

Kumar, R. R.; Mishra, S.; Kumar, C. (2017): Prioritizing the solution of cloud service selection using integrated MCDM methods under Fuzzy environment. *Journal of Supercomputing*, vol. 73, no. 11, pp. 4652-4682. DOI 10.1007/s11227-017-2039-1.

Kwok, P. K.; Lau, H. Y. (2019): Hotel selection using a modified TOPSIS-based decision support algorithm. *Decision Support Systems*, vol. 120, pp. 95-105. DOI 10.1016/j.dss.2019.02.004.

Li, D.; Zeng, W.; Yin, Q. (2018): Novel ranking method of interval numbers based on the Boolean matrix. *Soft Computing*, vol. 22, no. 12, pp. 4113-4122. DOI 10.1007/s00500-017-2625-4.

Lin, R. C.; Sir, M. Y.; Pasupathy, K. S. (2013): Multi-objective simulation optimization using data envelopment analysis and genetic algorithm: specific application to determining optimal resource levels in surgical services. *Omega*, vol. 41, no. 5, pp. 881-892. DOI 10.1016/j.omega.2012.11.003.

Liu, X.; Chu, J.; Yin, P.; Sun, J. (2017): Dea cross-efficiency evaluation considering undesirable output and ranking priority: a case study of eco-efficiency analysis of coal-fired power plants. *Journal of Cleaner Production*, vol. 142, pp. 877-885. DOI 10.1016/j.jclepro.2016.04.069.

Lotfi, F. H.; Fallahnejad, R. (2010): Imprecise Shannons entropy and multi attribute decision making. *Entropy*, vol. 12, no. 1, pp. 53-62. DOI 10.3390/e12010053.

Lotfi, F. H.; Jahanshahloo, G. R.; Ebrahimnejad, A.; Soltanifar, M.; Mansourzadeh, S. M. (2010): Target setting in the general combined-oriented CCR model using an interactive

MOLP method. *Journal of Computational and Applied Mathematics*, vol. 234, no. 1, pp. 1-9. DOI 10.1016/j.cam.2009.11.045.

Ma, H.; Hu, Z.; Li, K.; Zhu, H. (2019): Variation-aware cloud service selection via collaborative QoS prediction. *IEEE Transactions on Services Computing*. DOI 10.1109/TSC.2019.2895784.

Masdari, M.; Nabavi, S. S.; Ahmadi, V. (2016): An overview of virtual machine placement schemes in cloud computing. *Journal of Network and Computer Applications*, vol. 66, pp. 106-127. DOI 10.1016/j.jnca.2016.01.011.

Montgomery, D. C.; Peck, E. A.; Vining, G. G. (2015): *Introduction to Linear Regression Analysis*. Wiley, Hoboken.

Ouadah, A.; Hadjali, A.; Nader, F.; Benouaret, K. (2019): SEFAP: an efficient approach for ranking skyline web services. *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 2, pp. 709-725. DOI 10.1007/s12652-018-0721-7.

Özcan, E. C.; Ünlüsoy, S.; Eren, T. (2017): A combined goal programming-AHP approach supported with TOPSIS for maintenance strategy selection in hydroelectric power plants. *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 1410-1423. DOI 10.1016/j.rser.2017.04.039.

Pastor, J. T.; Ruiz, J. L.; Sirvent, I. (1999): An enhanced DEA Russell graph efficiency measure. *European Journal of Operational Research*, vol. 115, no. 3, pp. 596-607. DOI 10.1016/S0377-2217(98)00098-8.

Pérez, M. G. R.; Laprise, M.; Rey, E. (2018): Fostering sustainable urban renewal at the neighborhood scale with a spatial decision support system. *Sustainable Cities and Society*, vol. 38, pp. 440-451. DOI 10.1016/j.scs.2017.12.038.

Que, Y.; Zhong, W.; Chen, H.; Chen, X.; Ji, X. (2018): Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing. *International Journal of Advanced Manufacturing Technology*, vol. 96, no. 9-12, pp. 4455-4465. DOI 10.1007/s00170-018-1925-x.

Rai, D.; Kumar, P. (2016): Instance based multi criteria decision model for cloud service selection using TOPSIS and VIKOR. *International Journal of Computer Engineering and Technology*, vol. 7, no. 1, pp. 78-87.

Raju, K. S.; Kumar, D. N. (2006): Ranking irrigation planning alternatives using data envelopment analysis. *Water Resources Management*, vol. 20, no. 4, pp. 553-566. DOI 10.1007/s11269-006-3090-5.

Ramón, N.; Ruiz, J. L.; Sirvent, I. (2014): Dominance relations and ranking of units by using interval number ordering with cross-efficiency intervals. *Journal of the Operational Research Society*, vol. 65, no. 9, pp. 1336-1343. DOI 10.1057/jors.2013.90.

Rehman, Z.; Hussain, O. K.; Hussain, F. K. (2014): Parallel cloud service selection and ranking based on QoS history. *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 820-852. DOI 10.1007/s10766-013-0276-3.

Santos, J.; Bressi, S.; Cerezo, V.; Presti, D. L. (2019): SUP&R DSS: a sustainability-based decision support system for road pavements. *Journal of Cleaner Production*, vol. 206, pp. 524-540. DOI 10.1016/j.jclepro.2018.08.308.

Seghir, F.; Khababa, A. (2018): A hybrid approach using genetic and fruit fly optimization algorithms for QoS-aware cloud service composition. *Journal of Intelligent Manufacturing*, vol. 29, no. 8, pp. 1773-1792. DOI 10.1007/s10845-016-1215-0.

Seiford, L. M.; Zhu, J. (2002): Modeling undesirable factors in efficiency evaluation. *European Journal of Operational Research*, vol. 142, no. 1, pp. 16-20. DOI 10.1016/S0377-2217(01)00293-4.

Sengupta, A.; Pal, T. K. (2000): On comparing interval numbers. *European Journal of Operational Research*, vol. 127, no. 1, pp. 28-43. DOI 10.1016/S0377-2217(99)00319-7.

Serrai, W.; Abdelli, A.; Mokdad, L.; Hammal, Y. (2017): Towards an efficient and a more accurate web service selection using mcdm methods. *Journal of computational science*, vol. 22, pp. 253-267. DOI 10.1016/j.jocs.2017.05.024.

Serrai, W.; Abdelli, A.; Mokdad, L.; Serrai, A. (2019): How to deal with QoS value constraints in MCDM based web service selection. *Concurrency and Computation: Practice and Experience*, vol. 31, no. 24, pp. e4512. DOI 10.1002/cpe.4512.

Serrano, D.; Bouchenak, S.; Kouki, Y.; de Oliveira Jr., F. A.; Ledoux, T. et al. (2016): SLA guarantees for cloud services. *Future Generation Computer Systems*, vol. 54, pp. 233-246. DOI 10.1016/j.future.2015.03.018.

Shannon, C. E. (1948): A mathematical theory of communication. *Bell System Technical Journal*, vol. 27, no. 3, pp. 379-423. DOI 10.1002/j.1538-7305.1948.tb01338.x.

Sharma, Y.; Javadi, B.; Si, W.; Sun, D. (2016): Reliability and energy efficiency in cloud computing systems: survey and taxonomy. *Journal of Network and Computer Applications*, vol. 74, pp. 66-85. DOI 10.1016/j.jnca.2016.08.010.

Shemshadi, A.; Shirazi, H.; Toreihi, M.; Tarokh, M. J. (2011): A fuzzy VIKOR method for supplier selection based on entropy measure for objective weighting. *Expert Systems with Applications*, vol. 38, no. 10, pp. 12160-12167. DOI 10.1016/j.eswa.2011.03.027.

Shetty, J.; DMello, D. A. (2015). Quality of service driven cloud service ranking and selection algorithm using rembrandt approach. *International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials*. IEEE, pp. 126-132.

Silas, S.; Rajsingh, E. B.; Ezra, K. (2012): Efficient service selection middleware using ELECTRE methodology for cloud environments. *Information Technology Journal*, vol. 11, no. 7, pp. 868-875. DOI 10.3923/itj.2012.868.875.

Suciu, M.; Pallez, D.; Cremene, M.; Dumitrescu, D. (2013): Adaptive MOEA/D for QoS-based web service composition. *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 73-84. Springer, New York.

Sun, L.; Ma, J.; Zhang, Y.; Dong, H.; Hussain, F. K. (2016): Cloud-FuSeR: fuzzy ontology and MCDM based cloud service selection. *Future Generation Computer Systems*, vol. 57, pp. 42-55. DOI 10.1016/j.future.2015.11.025.

Sun, R.; Zhang, B.; Liu, T. (2016): Ranking web service for high quality by applying improved entropy-topsis method. *17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 249-254. IEEE.

Tchernykh, A.; Schwiegelsohn, U.; Talbi, E. G; Babenko, M. (2019): Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. *Journal of Computational Science*, vol. 36, pp. 100581. DOI 10.1016/j.jocs.2016.11.011.

Tone, K. (2001): A slacks-based measure of efficiency in data envelopment analysis. *European Journal of Operational Research*, vol. 130, no. 3, pp. 498-509. DOI 10.1016/S0377-2217(99)00407-5.

Vazirani, V. V. (2013): *Approximation Algorithms*. Springer Science & Business Media, New York.

Wang, S.; Liu, Z.; Sun, Q.; Zou, H.; Yang, F. (2014): Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing. *Journal of Intelligent Manufacturing*, vol. 25, no. 2, pp. 283-291. DOI 10.1007/s10845-012-0661-6.

Wang, S.; Zheng, Z.; Sun, Q.; Zou, H.; Yang, F. (2011): Reliable web service selection via QoS uncertainty computing. *International Journal of Web and Grid Services*, vol. 7, no. 4, pp. 410-426. DOI 10.1504/IJWGS.2011.044696.

Wang, Y. M.; Chin, K. S.; Yang, J. B. (2007): Measuring the performances of decision-making units using geometric average efficiency. *Journal of the Operational Research Society*, vol. 58, no. 7, pp. 929-937. DOI 10.1057/palgrave.jors.2602205.

Wang, Y. M.; Greatbanks, R.; Yang, J. B. (2005): Interval efficiency assessment using data envelopment analysis. *Fuzzy Sets and Systems*, vol. 153, no. 3, pp. 347-370. DOI 10.1016/j.fss.2004.12.011.

Wei, G.; Vasilakos, A. V.; Zheng, Y.; Xiong, N. (2010): A game-theoretic method of fair resource allocation for cloud computing services. *Journal of Supercomputing*, vol. 54, no. 2, pp. 252-269. DOI 10.1007/s11227-009-0318-1.

Wu, J.; Sun, J.; Song, M.; Liang, L. (2013): A ranking method for DMUs with interval data based on DEA cross-efficiency evaluation and TOPSIS. *Journal of Systems Science and Systems Engineering*, vol. 22, no. 2, pp. 191-201. DOI 10.1007/s11518-013-5216-7.

Wu, Z.; Ni, Z.; Gu, L.; Liu, X. (2010): A revised discrete particle swarm optimization for cloud workflow scheduling. *International Conference on Computational Intelligence and Security*, pp. 184-188. Nanning, China: IEEE.

Xu, C.; Ma, Y.; Wang, X. (2015): A non-parametric data envelopment analysis approach for cloud services evaluation. *Service-Oriented Computing-ICSOC 2014 Workshops*, pp. 250-255. Springer, Paris.

Zeng, L.; Benatallah, B.; Ngu, A. H.; Dumas, M.; Kalagnanam, J. et al. (2004): QoS-aware middleware for web services composition. *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311-327. DOI 10.1109/TSE.2004.11.

Zhang, J.; Gong, J.; Lin, H.; Wang, G.; Huang, J. et al. (2007): Design and development of distributed virtual geographic environment system based on web services. *Information Sciences*, vol. 177, no. 19, pp. 3968-3980. DOI 10.1016/j.ins.2007.02.049.

Zhang, X.; Lee, C.; Helal, S. (2019): iPOJO flow: a declarative service workflow architecture for ubiquitous cloud applications. *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1483-1494. DOI 10.1007/s12652-018-0792-5.

Zheng, Z.; Wu, X.; Zhang, Y.; Lyu, M. R.; Wang, J. (2012): QoS ranking prediction for cloud services. *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1213-1222. DOI 10.1109/TPDS.2012.285.