

Parallel Cloud Service Selection and Ranking Based on QoS History

Zia ur Rehman · Omar Khadeer
Hussain · Farookh Khadeer Hussain

Received: date / Accepted: date

Abstract The growing number of cloud services has made service selection a challenging decision-making problem by offering wide ranging choices for cloud service consumers. This necessitates the use of formal decision making methodologies to assist a decision maker in selecting the service that best fulfills the user's requirements. In this paper, we present a cloud service selection methodology that utilizes QoS history of cloud services over different time periods and performs parallel Multi-Criteria Decision Analysis to rank all cloud services in each time period in accordance with user preferences before aggregating the results to determine the overall rank of all the available options for cloud service selection. This methodology assists the cloud service user to select the best possible available service according to the requirements. The MCDM processes used for each time period are independent of the other time periods and are executed in parallel.

Zia ur Rehman
School of Information Systems
Curtin University, Perth, Australia
E-mail: zia-ur-rehman@postgrad.curtin.edu.au

Omar Khadeer Hussain
School of Information Systems
Curtin University, Perth, Australia
E-mail: o.hussain@cbs.curtin.edu.au

Farookh Khadeer Hussain
Decision Support and e-Service Intelligence Lab (DeSI Lab)
Quantum Computation and Intelligent Systems
School of Software, University of Technology, Sydney
Sydney, Australia
E-mail: Farookh.Hussain@uts.edu.au

1 Introduction

Cloud computing has several business advantages over conventional computing paradigms [1] due to its agility and flexibility, which has not only motivated organizations to develop their new applications on the cloud but also to migrate their existing business applications onto the cloud. To take maximum advantage of the full potential of cloud computing, a key issue for cloud services users is to ensure that the specific requirements and characteristics of their applications can be met by cloud service providers [2]. With the rapid growth of cloud computing, a number of service providers have appeared who offer similar services at various prices and performance levels. As a result of the dynamic nature of cloud services, which is a product of the elasticity and on-demand provision of computing resources, there are considerable fluctuations in the Quality of Service (QoS) levels of each service [3]. Therefore, capturing all the variety and inconsistency of service performance and selecting the right service according to each user's criteria are important tasks.

Existing approaches in the literature that assist service users in the decision making process of selecting a cloud service provider only consider the real-time QoS performance or average historical QoS performance of services. Such mechanisms may recommend a particular service, but that service may not be the most appropriate. The former approach (considering the real-time QoS performance) may lead to the selection of a service at local maxima because it ignores past QoS performance, while the latter method (considering the average historical QoS performance) does not capture the frequent variation in the QoS performance of cloud services. There is therefore a need for a cloud service selection approach that takes into account the multitude of available cloud services, variations in QoS performance (as well as price), and the user's criteria to rank available cloud services, and then assists in selecting the best and most advantageous service.

In this paper, we present such an approach for IaaS cloud service selection in which the top ranking services according to users' criteria are determined in different time slots (defined as non-overlapping periods of time), using a Multi-Criteria Decision Making (MCDM) method. The MCDM process in a time slot is independent of other time slots and is executed in parallel. These individual service selection results are then combined using an aggregation method to yield the overall service rank in the total time period, which is subsequently used to select the best service. Any MCDM method can be used to rank the services in this approach; however, we have used the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) and ELimination Et Choix Traduisant la REalité (Elimination and Choice Expressing Reality or ELECTRE). TOPSIS was proposed by Hwang and Yoon in 1981 [4], whereby the services ('alternatives' in MCDM terminology) are ranked on the basis of the Euclidean distance of an alternative (service) from the ideal and anti-ideal solutions. The service that is closest to the ideal solution and farthest from the anti-ideal solution achieves the highest rank and is therefore selected. ELECTRE was developed by Bernard Roy during 1960's as an outranking MCDM

method which determines the pairwise dominance relationship between the alternatives.

The remainder of the paper is organized as follows. In the next section, we discuss the related work in the area of cloud service selection and briefly discuss the role of MCDM methodologies. In Section 3, we present our overall framework for cloud service selection that assists a service user to decide on the most appropriate service from the services available by ranking the latter using a parallelly executed MCDM process. In Section 4, we present our approach for cloud service selection, followed by the experimental validation of the proposed approach in Section 5. Section 6 concludes the paper.

2 Related Work

2.1 Cloud Service Selection

A number of research works dealing with the issue of cloud service selection have been published in recent years. In this section we present an overview of some of these research efforts.

Pastaki Rad et al. [5] presented a general survey and comparison of prominent cloud platforms by leading cloud providers with an emphasis on the key differentiating features of each platform. Peng et al. [6] provided a general survey of popular cloud middle-ware, such as Eucalyptus, NIMBUS and Open Nebula, and discussed their architecture, characteristics and application. A virtual machine image selection service for cloud computing environments has been proposed by Filepp et al. [7]. This proposed image selection service maintains a repository of image configuration details and employs an algorithm to order the images based on conformance with specified user requirements and policies by best-fit and least-cost optimization. Li et al. [8, 9, 10] discussed the problem of comparing different cloud services and identified the basic attributes for each type of cloud service that must be taken into consideration when comparing one cloud service with another. They also differentiated between the performance of a cloud service itself and the performance of an application deployed on that cloud [11]. Nie et al. [12] presented a complete evaluation index system of cloud services and utilized Analytical Hierarchy Process (AHP) to calculate the weights of attributes for service evaluation. They also established a number of qualitative models for purchase decision making.

A set of measurement indexes for comparing different cloud services, called the *Service Measurement Index (SMI)*, has been devised and is based on common characteristics of cloud services identified by the Cloud Service Measurement Index Consortium (CSMIC) [13]. Garg et al. [2, 14] proposed a framework – called *SMICloud* – for comparing and ranking cloud services on the basis of SMI criteria. The proposed framework systematically measures all the QoS attributes in SMI and then uses an AHP-based mechanism to rank the cloud services. Han et al. [15] proposed a cloud service recommender system for the

cloud market that helps a user to select the best combination of services from different cloud providers by matching the specific requirements of the user with a suitable cloud service. This system maintains a resource register to keep a record of all the available resources in the cloud market and uses this information to rank and calculate the QoS values of services. They also outline the ranking methods for each type of cloud service (SaaS, IaaS etc.).

Kang and Sim [16, 17, 18] developed a cloud service search engine called *Cloudle*, which is based on a cloud ontology consisting of cloud concepts, individuals of those concepts and their mutual relationships. All services are registered in a database and a query processor executes the user's query, which is sent to a similarity reason engine that performs similarity reasoning between the query and the concepts in the database using cloud ontology. The output of the *Cloudle* search engine is an ordered list of cloud services. The services are ordered on the basis of three criteria (1) concept similarity, (2) price utility, and (3) cost utility. Chen et al. [19] presented a framework that enables automatic conflict detection between the user's criteria and enterprise policies in cloud service selection for enterprises. This system aims to tackle the difficulties of cloud service selection with an emphasis on the involvement of enterprise policies. It checks various conflicts that result from the violation of enterprise policies and inconsistency in a cloud service user's requirements. This check is followed by the selection of an appropriate service that satisfies the user's requirements and also complies with enterprise policies, using constraint programming. Wang et al. [20] proposed a QoS evaluation methodology for service oriented cloud computing using fuzzy synthetic decision making according to cloud users' preferences and calculating the uncertainty of cloud services by applying a cloud model on the monitored cloud QoS data. Zeng et al. [21] developed a cloud service selection algorithm that uses a service discoverer to find all the available services and then processes the cloud service user's request by employing a maximized-gain and minimized-cost service selection algorithm. This algorithm aggregates the gain and cost values by a weighted sum of both types of values (where weights represent the relative importance of each value). Godse and Mulik [22] proposed an approach for selecting SaaS products. They argued that to make an informed decision, it is necessary to have quantifiable values instead of subjective opinions. They proposed several key factors – such as functionality, architecture, usability, vendor reputation and cost – for SaaS selection and used AHP for service selection decision making. In one of our earlier papers [23], we presented a framework for a user feedback-based cloud service monitoring system which collects feedback related to the QoS performance of cloud services from existing cloud service users and maintains a repository of this information which can be used by service selection mechanisms to recommend appropriate cloud services to users. In another paper [24], we presented the cloud service selection problem as a multi-criteria decision making problem by proposing a mathematical framework for multi-criteria cloud service selection.

To summarize, as shown in Table 1, there are a variety of approaches proposed in the literature, several of which are based on MCDM techniques, that

assist a user in making a service selection decision in the Cloud environment. None of the existing approaches, however, simultaneously consider the QoS history and the frequent variations therein during the service selection process, and they are therefore unable to capture these important factors which, as discussed in the previous section, are necessary to ensure accurate service selection.

2.2 MCDM in Cloud Service Selection and Problem Definition

Multi-criteria decision making (also referred to as multi-criteria decision analysis (MCDA)) is a collection of methodologies for comparing, ranking and selecting multiple alternatives with multiple attributes [25]. MCDM techniques are extensively used in decision support systems [26–30]. MCDM is used in situations where several alternatives are present and a decision has to be made in favor of one alternative on the basis of involving more than one criterion. Such situations often arise in real world problems where decisions have to be made in the presence of multiple conflicting criteria for judging available alternatives and where making compromises or trade-offs related to outcomes is necessary. It happens quite often that one alternative is better than others on the basis of one or more criteria, while the same alternative is the worst when judged on the basis of other criteria. Several MCDM methodologies have been developed in the literature but all are based on three basic working principles, namely: (1) Multi-attribute Utility Theory (MAUT)(2) Outranking methods and (3) hierarchical and network-based methods. There are several methodologies in each of these categories. The notable methods based on MAUT are: Min–Max, Max–Min and TOPSIS. The outranking methods include the ELECTRE and PROMETHEE, each of which has several variants. The AHP is a hierarchical method, while ANP is a network-based method, and thus both fall into the third category of MCDM methods. The typical properties of MCDM problems as outlined by [31] and [32] are analogous to the cloud service selection problem and underpin the notion of a MCDM-based cloud service selection mechanism, and the problem of cloud service selection falls into the category of multi-criteria selection problems. Comparison between the available cloud services according to the variability in their performance over time is necessary to generate a ranking of the cloud services for cloud service selection. Since cloud services have numerous characteristics, all of these characteristics need be considered in the comparison of any two clouds. However, comparison between two services is not trivial because one cloud service may be better in terms of some characteristics, while another service may excel in other attributes. Furthermore, the characteristics of cloud services may not be equally important for fulfilling specific user requirements in all the time periods over which the decision has to be made. In such situations Multi-Criteria Decision Making (MCDM) techniques are useful for the comparison and ranking of cloud services.

Publication	Area discussed	Summary	QoS based	Variation in QoS with time
Pastaki Rad et al. [5]	Cloud service platforms	General survey of cloud service platforms and their key features.		
Peng et al. [6]	Cloud middleware	General survey of popular cloud middleware.		
Filepp et al. [7]	Virtual machine image selection	Selects virtual machine images using a image configuration repository and minimum cost maximum gain algorithm		
Li et al. [10]	Cloud service comparison	Highlights the problems in comparing different clouds and identifies basic attributes of each type of cloud for comparison		
Li et al. [11]	Cloud performance	Difference between cloud performance and cloud application performance		
Nie et al. [12]	Cloud service selection	Evaluation index system for cloud service purchase decision making using AHP.	no	no
Siegel and Perdue [13]	Cloud service comparison	A set of measurement indexes is proposed for comparing different cloud services.		
Garg et al. [2, 14]	Cloud comparison and ranking	A framework for measuring the QoS attributes and AHP based ranking of cloud services	yes	no
Han et al. [15]	Cloud service composition recommender	A system aimed at helping the user in selecting the best combination of service from different cloud providers by matching the users requirements with QoS values of services.	yes	no
Kang and Sim [16, 17, 18] Kang and Sim	Ontology based cloud service search engine	An ontology based cloud service search engine that maintains a database to register the available cloud services and user query is processed responded by presenting an ordered list of the available services. The list is ordered on the basis of concept similarity, price utility and cost utility.	no	no
Chen et al. [19]	Cloud service selection	Detects conflicts in user's requirement and enterprise policies and then selects service using constraint programming	no	no
Wang et al. [20]	Cloud service selection	QoS evaluation using fuzzy synthetic decision making based users preferences and a measurement of uncertainty of cloud services by applying a cloud model on monitored cloud data.	yes	no
Zeng et al. [21]	Cloud Service selection	Uses the maximum-gain and minimum-cost algorithm for cloud service selection	no	no
Godse and Mulik [22]	Cloud service selection	SaaS selection using AHP	no	no
Rehman et al. [23]	Cloud service Monitoring	User feedback to measure cloud service QoS		
Rehman et al. [24]	Cloud service selection	A framework for MCDM based cloud service selection	no	no
Zheng et al. [3]	QoS Ranking prediction	A cloud QoS ranking prediction framework based on collaborative filtering recommender system theory.	yes	no

Table 1 Summary of related literature on cloud service selection

To summarize, a variety of approaches to cloud service selection have been proposed in the literature, according to different factors and using a range of techniques. An unsupported factor that needs to be considered in such systems is the ability to capture the variability in QoS and the dynamic nature of cloud environments in the process of cloud service selection. Most existing approaches fail to consider this aspect, hence the services they select may not capture such variations in their decisions. To achieve this goal, we propose an approach for cloud service management in which we divide the total period of QoS performance history over which service management decisions have to be made into two parts, namely, pre-interaction start phase (pre-interaction) and post-interaction start phase (post-interaction) (Figure-1). Time spot is defined as that instance of time at which the service selection decision is to be made. Pre-interaction time period is that period of time before the time spot in which the past QoS performance of each service is analyzed to select the most appropriate service. Post-interaction time period is that period of time after the time spot in which the real-time QoS performance of the selected service and other available services is monitored and analyzed to ensure that the needs of the user are being fully achieved, and, if they are not, to recommend service migration if another service can fulfill the user's needs at lower cost. In this paper, our aim is to assist a user to make an informed decision in selecting the most capable service; therefore, we focus only on the pre-interaction phase time period.

In the next section we describe our integrated framework for cloud service selection and its constituent parts.

3 Framework for cloud service selection

We propose a cloud service selection framework (Figure-2) which relies on integrated QoS information – collected from multiple sources – for service selection decision making. The sources of information include: (1) service specification published by the service providers, (2) cloud service monitoring, and

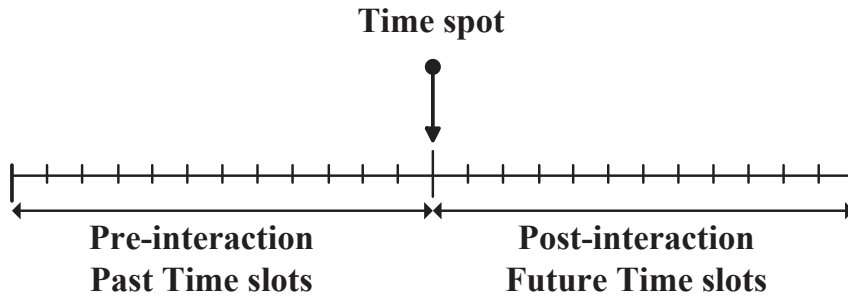


Fig. 1 The pre-interaction and post-interaction time periods.

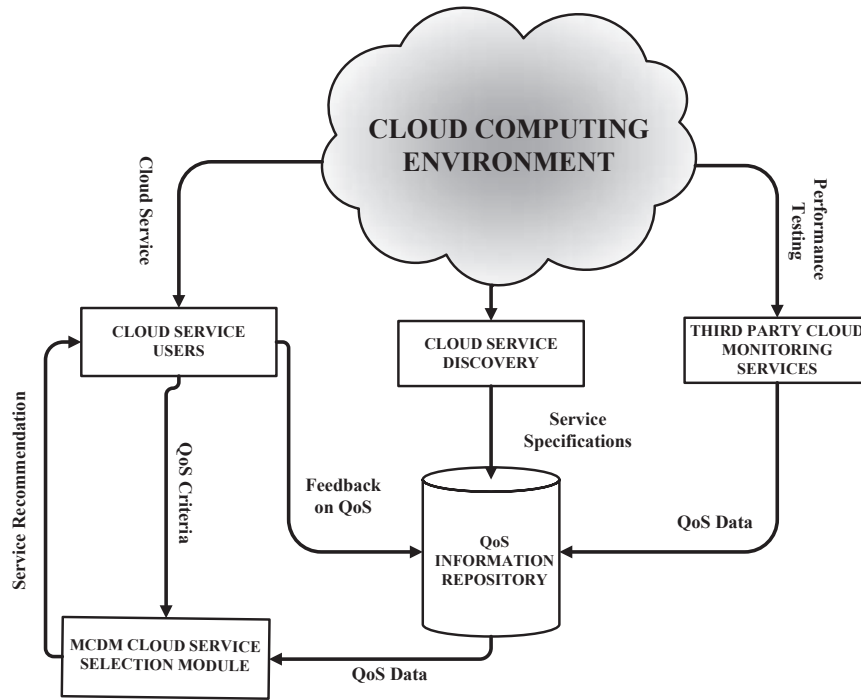


Fig. 2 Flow of information between different modules in cloud service selection.

(3) feedback from existing cloud service users. This framework consists of several modules: (1) cloud service discovery, (2) cloud service monitoring, (3) QoS information repository, and (4) MCDM cloud service selection module. The cloud services available in the cloud environment are searched by a service discovery module and their specifications are stored in the QoS repository which serves as a register of available cloud services in addition to having the function of storing QoS information. The registered cloud services are monitored by a cloud service monitoring module which executes benchmarks tests on the available cloud services and the collected data is stored in the QoS repository. In addition to this source of QoS information, existing cloud service users also provide QoS information about the service they use. The QoS repository is a record of the QoS of available services and this information is used by the decision making module to recommend appropriate services to new users.

Cloud service discovery: This module searches the cloud environment for available cloud services and their specifications and also acts as an interface between the framework and the cloud environment by collecting the service specification information published by cloud providers. In addition to looking for available new services, this module also keeps track of changes to the specifications of existing services.

Cloud service monitoring: This module monitors the services registered in the cloud service repository and collects data on the QoS of the available services by executing a benchmark test on the available services as well as using the data collected by third party cloud monitoring services.

QoS information repository: This module stores the data collected by the service discovery and the service monitoring modules. It also stores QoS information received from existing cloud service users.

MCDM cloud service selection module: This module obtains the QoS information contained in the QoS repository and the criteria preference values from the user and performs multi-criteria decision analysis on this information to rank the available services.

In the next section, we present the detailed service selection approach employed in the MCDM cloud service selection module.

4 Proposed Approach for MCDM in the Pre-Interaction Phase

In our approach, a long term QoS history of available services is utilized for decision analysis, unlike some previous cloud service decision making approaches which are driven by QoS performance at one instance of time, or by the average QoS. Currently there are various cloud QoS monitoring services that monitor and store the long-term QoS history of available services. Our aim is to use the QoS performance and price history of available cloud services to select the most appropriate service, avoiding the selection of a service at local maxima (which happens if the real-time QoS data of only the current time is used) but without entirely losing the information about variations in QoS performance (which happens when only the average QoS is used). Our proposed approach is depicted in Figure-3, and involves the following key steps:

- Step A: To capture the variations in QoS over time, we divide the pre-interaction time period for cloud service management into a number of equal non-overlapping time slots (Figure 1). The criteria $C_1, C_2 \dots C_n$ for service selection are identified by the user and in each time slot the QoS performance of all the services measured on the basis of the identified criteria is retrieved by the MCDM module from the QoS information repository (Figure 2).
- Step B: The identified QoS criteria are not equally important for users in decision making. Each user has specific preferences regarding the relative importance of individual criteria. This important information is expressed in the form of criteria weights i.e. $\{w_{c_1}, w_{c_2} \dots w_{c_n}\}$, where each criterion C_i has a weight w_{c_i} .
- Step C: The QoS performance data of all the available services in each time slot form a decision matrix that is used with the criteria weights to find the best service by employing a MCDM technique. The MCDM method is parallelly applied to all time slots to compute the service rank of each service, and the top ranking service in each time slot is selected.
- Step D: To consider the dynamic nature of time when selecting a service, we consider the freshness of the QoS values of a service depending upon its

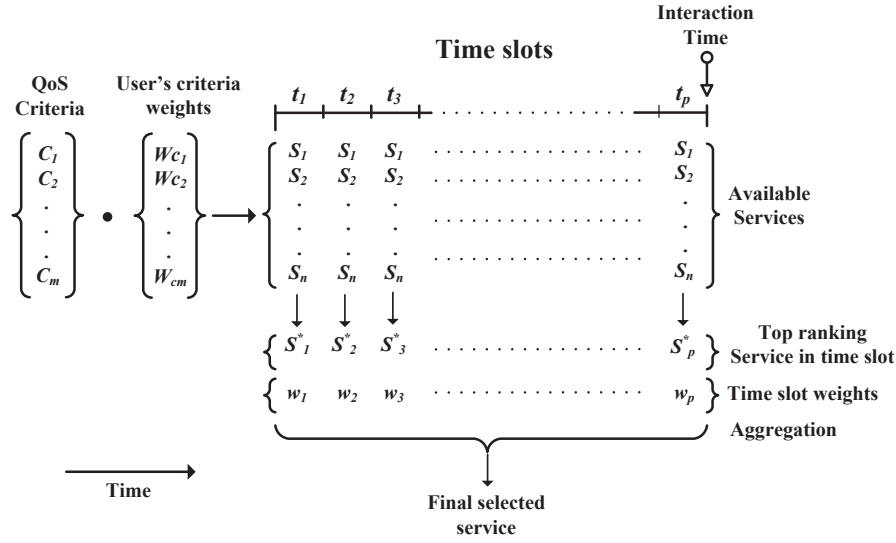


Fig. 3 Overview of the proposed approach for service selection based on time decay and QoS performance of services in different time slots

distance from the time spot at which the decision has to be made. Each time slot is therefore assigned a time slot weight which progressively decreases from a maximum value of 1.0 (for the most recent time slot with respect to the time spot) to successively lower values for older time slots until it reaches a minimum value of 0.4. Thus the QoS performance values of services in recent time slots have a much higher impact on the final service selection decision than the values of services in older time slots.

Step E: The service selection results obtained in Step C above are combined by an aggregation process using the time slot weights determined in Step D. The aggregation yields the overall service rank in the pre-interaction time period, from which the final service selection decision is made.

The sequence of flow in the working of our proposed approach is as shown in Figure 4. We elaborate Step C of our proposed approach in the following section, while Steps D and E are discussed in sections 4.2 and 4.3 respectively.

4.1 Finding the top ranked service in each time slot

The objective of this step is to find the highest ranked service in each time slot based on the QoS performance values of the available services, using MCDM. We use two MCDM techniques; TOPSIS and ELECTRE as given below. This step is performed in parallel for each time slot under consideration.

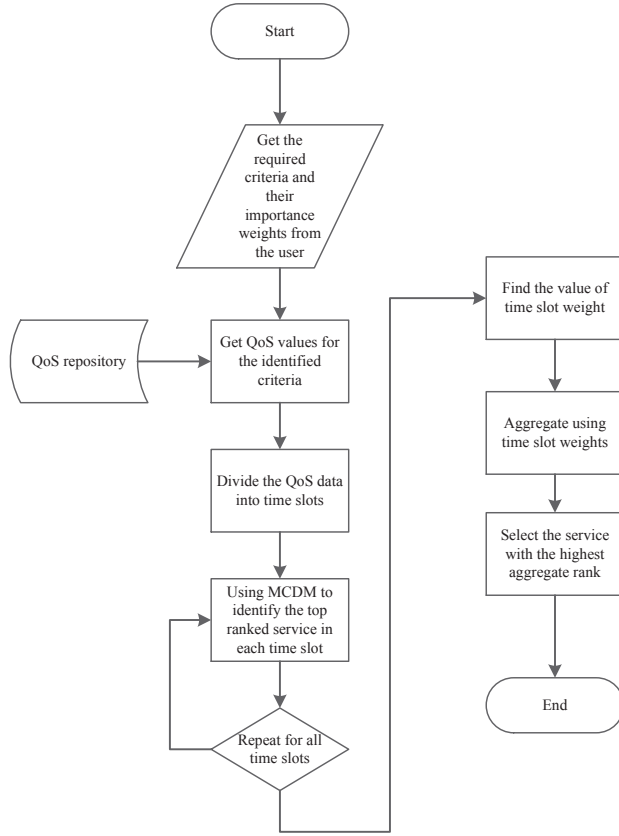


Fig. 4 Flowchart showing the sequence of steps in the proposed approach

4.1.1 TOPSIS Method

The calculation steps for determining the service ranks in an individual time slot by TOPSIS are as follows:

Step 1: QoS values of all the services in each time slot form an evaluation matrix D , which has the following form.

$$D = \begin{matrix} & C_1 & C_2 & \dots & C_n \\ \begin{matrix} S_1 \\ S_2 \\ \vdots \\ S_m \end{matrix} & \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{pmatrix} \end{matrix} \quad (1)$$

Where, $S_1, S_2 \dots S_m$ are the m available services; $C_1, C_2 \dots C_n$ are the n criteria and each r_{ij} is a measurement of the performance of service S_i under criterion C_j .

Step 2: Since each criterion has its own units and range, the evaluation matrix in Equation-1 is normalized to make the QoS values of different criteria comparable. The normalized evaluation matrix N is given by:

$$N = \begin{pmatrix} n_{11} & n_{12} & \dots & n_{1n} \\ n_{21} & n_{22} & \dots & n_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ n_{m1} & n_{m2} & \dots & n_{mn} \end{pmatrix} \quad (2)$$

$$\text{Where } n_{ij} = \frac{r_{ij}}{\sqrt{\sum_{i=1}^m (r_{ij})^2}}$$

Step 3: The user's preference information is incorporated by finding the weighted evaluation matrix. If the criteria preference weights provided by the cloud service user ('*decision maker*' in MCDM terminology) are $w_{c_1}, w_{c_2}, \dots, w_{c_n}$ (such that: $w_{c_i} \geq 0$ and $\sum_{i=1}^n w_{c_i} = 1$), then the corresponding weight matrix is given by an $n \times n$ diagonal matrix W_c whose diagonal elements are $w_{c_1}, w_{c_2}, \dots, w_{c_n}$. The weighted evaluation matrix V is determined by the product of the normalized evaluation matrix N from Equation-2 and the diagonal weight matrix W_c , as shown in Equation-3 below.

$$\begin{aligned} V &= \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nn} \end{pmatrix} \\ &= \begin{pmatrix} n_{11} & n_{12} & \dots & n_{1n} \\ n_{21} & n_{22} & \dots & n_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ n_{m1} & n_{m2} & \dots & n_{mn} \end{pmatrix} \begin{pmatrix} w_{c_1} & 0 & \dots & 0 \\ 0 & w_{c_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{c_n} \end{pmatrix} \end{aligned} \quad (3)$$

Where, $w_{c_i} \geq 0$ and $\sum w_{c_i} = 1$.

Step 4: The weighted normalized decision matrix V is used to determine the ideal solution (A^*) and the anti-ideal solution (A') as follows:

$$A^* = \{v_j^*, j = 1, 2, \dots, k\} = \{\text{Max } q_{ij}, \forall i; j = 1, 2, \dots, 3\} \quad (4)$$

$$A' = \{v_{*j}, j = 1, 2, \dots, k\} = \{\text{Min } q_{ij}, \forall i; j = 1, 2, \dots, 3\} \quad (5)$$

Step 5: The separation measure for each service from the ideal solution (denoted by D_i^*) and the anti-ideal solution (denoted by D_i') are determined by:

$$D_i^* = \left[\sum_j (v_{ij} - v_i^*)^2 \right]^{\frac{1}{2}} \quad (6)$$

and

$$D'_i = \left[\sum_j (v_{ij} - v'_i)^2 \right]^{\frac{1}{2}} \quad (7)$$

Step 6: The final step in TOPSIS is to find the similarity index which combines the two separation measures obtained in the previous step. The similarity index G_i corresponding to each service S_i is given by:

$$G_i = \frac{D'_i}{D'_i + D_i^*} \quad (8)$$

The service corresponding to the highest G_i is selected as the best service within the time slot under consideration.

4.1.2 ELECTRE Method

Compared with MAUT-based methods such as TOPSIS, this method is more complicated; the simplest variant of ELECTRE involves up to 10 steps. It performs a pairwise comparison between the alternatives and builds an out-ranking relationship between them. This relationship is then used to identify and eliminate the alternatives that are dominated by other alternatives to yield a smaller set of alternatives (called the kernel). A variant of this technique called ELECTRE II yields a complete rank order of the original set.

The first three steps of this method are similar to the TOPSIS method outlined in Section 4.1.1. The remaining steps after calculating the normalized decision matrix V (Equation-3) are as follows:

Step 4: Let $J = \{j | j = 1, 2, \dots, n\}$ be the set of criteria and concordance sets $S_{k,l}$ and discordance sets $D_{k,l}$ for all pairs A_k and A_l of alternatives. Where $k, l = 1, 2, \dots, m$ and $l \neq k$. Also,

$$S_{kl} = \{j | r_{kj} \geq r_{lj}\} \quad (9)$$

and

$$D_{kl} = \{j | r_{kj} \leq r_{lj}\} = J - S_{kl} \text{ or } D_{kl} = S_{kl}^c \quad (10)$$

Step 5: Find the concordance matrix:

$$I = \begin{pmatrix} - & i_{12} & i_{13} & \dots & i_{1m} \\ i_{21} & - & i_{23} & \dots & i_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ i_{m1} & i_{m2} & \dots & i_{m,(m-1)} & - \end{pmatrix} \quad (11)$$

where i_{lk} is the concordance index for the alternative pair A_k and A_l and is given by:

$$i_{kl} = \sum_{j \in S_{k,l}} w_j \quad ; \quad \sum_{j=1}^n w_j = 1$$

Step 6: Find the discordance matrix:

$$NI = \begin{pmatrix} - & ni_{12} & ni_{13} & \dots & ni_{1m} \\ ni_{21} & - & ni_{23} & \dots & ni_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ ni_{m1} & ni_{m2} & \dots & ni_{m,(m-1)} & - \end{pmatrix} \quad (12)$$

$$\text{Where } ni_{k,l} = \frac{\max_{j \in D_{k,l}} |v_{kl} - v_{lj}|}{\max_{j \in J} |v_{kl} - v_{lj}|}$$

Step 7: Calculate the arithmetic mean of the concordance matrix, given by:

$$\bar{I} = \sum_{k=1}^m \sum_{l=1}^m \frac{i_{k,l}}{m(m-1)} \quad (13)$$

Using the above calculated \bar{I} find the Boolean matrix F, i.e.

$$F = \begin{pmatrix} - & g_{12} & f_{13} & \dots & f_{1m} \\ f_{21} & - & f_{23} & \dots & f_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{m1} & f_{m2} & \dots & f_{m,(m-1)} & - \end{pmatrix} \quad (14)$$

Where,

$$\begin{aligned} f_{kl} &= 1; i \geq \bar{I} \\ &= 0; i \leq \bar{I} \end{aligned}$$

Step 8: Similarly calculate the arithmetic mean of the discordance matrix:

$$\bar{NI} = \sum_{k=1}^m \sum_{l=1}^m \frac{ni_{k,l}}{m(m-1)} \quad (15)$$

The corresponding Boolean matrix G for the discordance matrix is given by:

$$G = \begin{pmatrix} - & g_{12} & g_{13} & \dots & g_{1m} \\ g_{21} & - & g_{23} & \dots & g_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{m1} & g_{m2} & \dots & g_{m,(m-1)} & - \end{pmatrix} \quad (16)$$

Where,

$$\begin{aligned} g_{kl} &= 1; ni \leq \bar{NI} \\ &= 0; ni \geq \bar{NI} \end{aligned}$$

Step 9: Using matrices F and G, form the composite matrix H such that:

$$H = \begin{pmatrix} - & h_{12} & h_{13} & \dots & h_{1m} \\ h_{21} & - & h_{23} & \dots & h_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{m1} & h_{m2} & \dots & h_{m,(m-1)} & - \end{pmatrix} \quad (17)$$

Where, $h_{k,l} = f_{k,l} \cdot g_{k,l}$

Step 10: The matrix H indicates the preference such that $h_{k,l} = 1 \implies A_k \prec A_l$, but it is still possible that A_k is dominated by other alternatives. In our framework, we calculate the row sum of this matrix which gives the rank of each service, and the service corresponding to the highest rank is selected.

4.2 Calculation of Time slot Weights in the Pre-Interaction Phase

The objective of this step is to reflect the relative importance of time slots by assigning an appropriate weight to each time slot. As mentioned previously, in our approach we consider that time slots nearest to the time spot have more importance than the distant time slots (Figure-3). If there are n time slots $t_1, t_2 \dots t_n$, then the corresponding time slot weight for each time slot t_i is given by the following logistic decay function i.e.:

$$w_i = A + \frac{K - A}{(1 + e^{-B(\Delta t_i - M)})^{1/2}} \quad (18)$$

Where, Δt_i is the time interval between the interaction time spot t_p and the time slot in consideration t_i .

The properties of this logistic decay function are controlled by the constants A, K, B , and M . Where, A is the lower asymptote, K the upper asymptote, B the growth rate and M the time of maximum growth. This gives a weight to each time slot in such a way that the most recent time slots (which are immediately preceding the time spot) have a higher weight as compared to the distant time slots which will have a lower weight. In our approach, we consider that the first few time slots closest to the time spot have the maximum weight ($w_t \approx 1$); thereafter, the weight decreases for subsequent time slots and remains constant after reaching a minimum value of 0.4 (represented by the constant K in Equation-18). In Figure-5, we plot 3 decay curves, each varying on the importance of weights that it gives to the time slots nearest to the time spot. Curves 1, 2 and 3 give a weight of 1 to the 50, 100, and 150 time slots (value of M) from the time spot, respectively. The values of other constants for plotting these curves are $A = 1$; $K = 0.4$ and $B = 0.5$.

4.3 Aggregation of individual time slot results

After determining the top ranking service in each time slot using a MCDM technique (Step C) and calculating the weight (time decay) of each time slot

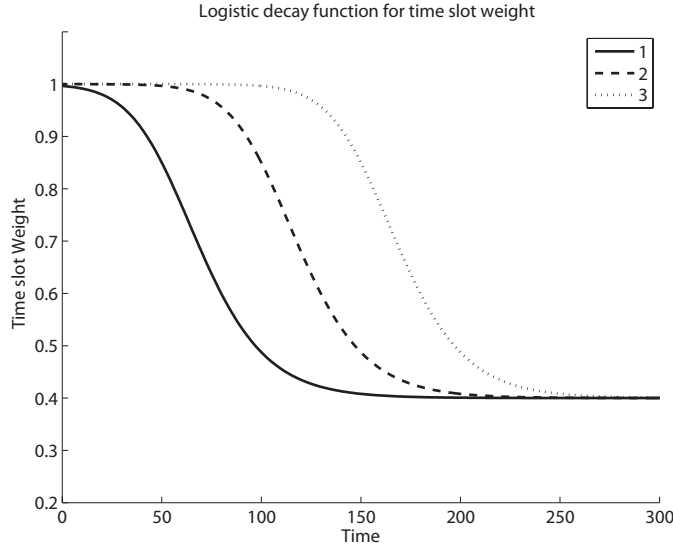


Fig. 5 Logistic decay functions for time slot weights

(Step D), the overall rank of a service in the entire pre-interaction period is calculated in this step. Using the individual service selection outcome for all time slots, we construct a Boolean matrix (Equation-19), such that the element u_{ij} corresponding to service S_i and time slot t_j equals 1 only if service S_i is the top ranked service in time slot t_j .

$$U = \begin{matrix} & t_1 & t_2 & \dots & t_n \\ \begin{matrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{matrix} & \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \dots & u_{mn} \end{pmatrix} \end{matrix} \quad (19)$$

$$\text{Where } u_{ij} = \begin{cases} 1 & \text{if } S_i \text{ ranks at the top in time slot } t_j \\ 0 & \text{otherwise} \end{cases}$$

Thus each column of the above matrix U represents the MCDM outcome for all available services in one time slot, while each row represents the TOP-SIS outcome for one service in all time slots. Using this matrix, the overall aggregated rank R_i of service S_i is calculated by

$$R_i = \sum_{j=1}^n w_j \cdot u_{ij} \quad (20)$$

Where w_j is the time slot weight

This process is repeated for all the available services (each row of the matrix U) to find the overall rank of each service in the entire pre-interaction period. Alternatively, the product of the Boolean matrix U and a column vector containing the time slot weights w_1, w_2, \dots, w_n , yields a column vector representing the overall service ranking, i.e.

$$\begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \dots & u_{mn} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \quad (21)$$

Where w_j , ($j = 1, 2 \dots n$), is the time slot weight and the service S_k corresponding to the maximum overall ranking R_k is then selected as the best service for the user.

In the next section, we discuss the experimental validation of our proposed approach for cloud service selection.

5 Experimental Validation

5.1 Data

To validate our approach we used the QoS monitoring data of five Amazon EC2 IaaS cloud services. The data was collected by *cloudclimate* (www.cloudclimate.com) using the PRTG monitoring service (<https://prtg.paessler.com>). The dataset consists of hourly measurements of response time for 300 days (from 1-26-2012, 2 PM to 21-11-2012, 2 PM) of the five EC2 instances to short load tests which reflect the CPU, Memory and I/O performance of the monitored services. In addition to these three criteria, we included the price per hour for each service, quoted by Amazon (www.amazon.com), as the fourth criterion. The EC2 services included in this dataset and their respective prices for hourly usage are given in Table 2.

Service	Detail	Instance Type	Cost(\$/hr.)
S_1	EC2 EU	small	0.0885
S_2	EC2 EU	micro	0.0200
S_3	EC2 SA	micro	0.0270
S_4	EC2 US East	small	0.0650
S_5	EC2 US West	micro	0.0250

Table 2 Amazon Services in the dataset

The services in this dataset were of EC2 small and micro instance type. We observed that, in terms of performance, the micro instance services overwhelmingly surpassed the small instance services. The performance of the

CPU, memory and disk of the micro instances – although more volatile – appears to be 3 to 5 times better than the performance of small instances. Our proposed approach relies on MCDM, therefore a dataset consisting of more than three services was necessary to test our approach. As no other real data were available for this experiment, we scaled the data using range scaling to make them comparable for this simulation while keeping intact the temporal QoS variations, rather than generating artificial data. QoS data for each service was scaled along all criteria over the entire dataset (i.e. all time slots) using the following formula,

$$scale(r_{ij}) = \frac{r_{ij}}{max(r_j) - min(r_j)} \times 1000 \quad (22)$$

Where, r_{ij} is QoS value of service S_i in terms of QoS criteria C_j and $max(r_j)$ and $min(r_j)$ are the maximum and minimum values, respectively, for each criterion (in column j of the decision matrix in Equation 1). We used a time slot length of 24 hours, dividing the available dataset into 300 time slots and using the QoS values of 2.00 PM each day as the decision matrix for each time slot. A portion of the data (for time slots 1 to 100) is given in Table 3, where C_1 , C_2 , and C_3 represent the QoS of CPU, memory and I/O respectively, while C_4 (not shown in Table 3) is the cost per hour for usage (shown in Table 2:Column-4), which was constant throughout the duration of the data collection and $S_1 - S_5$ represent the 5 services. The complete dataset is plotted in a graphical format in Figure-6, which shows continuous variation in the QoS criteria values. The arithmetic mean of the dataset being considered is given in Table 4. These values are used as input for our simulation models (described in the next subsection).

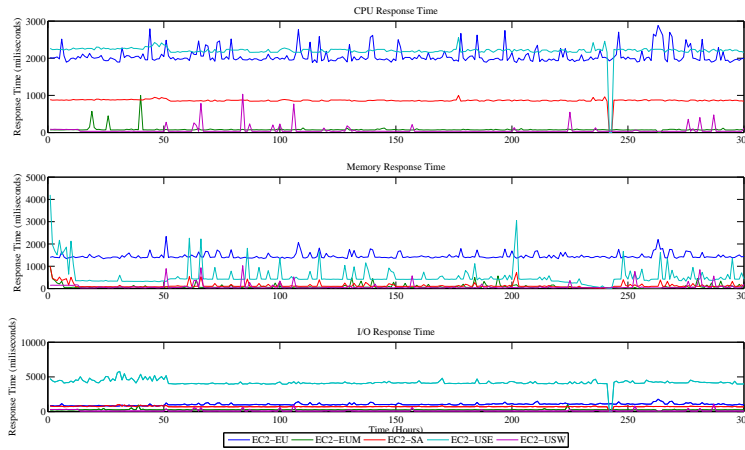


Fig. 6 Variation in QoS over time (days)

Services	Average Response Time (milliseconds)		
	CPU	Memory	I/O
S_1	2056.19	1455.72	1035.82
S_2	80.77	81.94	260.42
S_3	860.15	126.66	722.40
S_4	2200.70	532.28	4187.19
S_5	56.41	73.93	122.34

Table 4 Average QoS of the 300 time slots.

5.2 Simulation models

The dataset described in the previous sub-section was used to select the best service in four different simulation models. The four simulations were performed using TOPSIS and ELECTRE as the means for MCDM at each time slot. The objective was to discover whether there was any difference between service selection outcome using average QoS data over the pre-interaction period and service selection outcome using their individual rank in each time slot and also to determine the effect of time slot weights on the overall service ranking. The four simulation models used were:

Model I Service selection by applying MCDM to average QoS values (existing approaches).

Model II Service selection by aggregation of MCDM outcomes in each time slot and using constant criteria weights without time decay.

Model III Service selection by time decay aggregation of MCDM outcomes in each time slot and using constant criteria weights. Three variations of the logistic time decay function were used in this simulation.

Model IV Service selection with different criteria weights for each time slot, determined using the entropy method. In this simulation model, we repeat the experiments in simulation models I,II and III with entropy weights of each criterion as simulation models I^e II^e and III^e respectively.

In simulation models III and III^e , three logistic decay functions are used to calculate the weight of each time slot. These functions give a maximum value of 1 to the time slots near the time spot and logistically decrease the weight to the minimum value of 0.4 for older time slots. The first logistic decay function gives the maximum weight of 1 to the first 10 time slots from the time spot and then logistically decreases to 0.4 up to the 150th time slot. In the second decay function, the lowering of the time slot weight from the maximum value of 1 begins after a longer period of time from the time spot (from 50th time slot), thereby giving older time slots slightly more importance than the first decay function. In the third decay function, the weight decay starts after 100th time slot from the time spot and decreases to the minimum value of 0.4 up to the 160th time slot. Using the logistic decay function with three different parameters enables us to see the relative effect of the manner in which the time slot weight decay affects the final aggregated service selection.

In the simulation models I, II and III, we used neutral criteria weights (the same weights for all criteria). When average QoS is used for MCDM-based service selection, the criteria weights cannot reflect users' changing requirements over time. By contrast, our approach performs separate MCDM analysis for each time slot, therefore it is possible to use different criteria weights in different time slots. In simulation model IV we repeat the simulation model II and III by dynamically calculating the criteria weights using the entropy method for each time slot to demonstrate this additional capability of our approach.

The entropy method estimates the relative importance (weights) of the criteria using the concept of Entropy in information theory. The entropy value gives an estimate of the amount of information contained in the decision matrix (Equation 1) and is given by the following equation [33].

$$e_j = \frac{1}{\ln m} \sum_{i=1}^m r_{ij} \ln(r_{ij}), j \in [1, n] \quad (23)$$

where r_{ij} are the values in decision matrix (Equation 1) and $r_{ij} \cdot \ln r_{ij} = 0$ if $r_{ij} = 0$. Using these entropy values the weight for each criterion is calculated as;

$$w_{c_j} = \frac{1 - e_j}{\sum_{j=1}^n (1 - e_j)} \quad (24)$$

Using this method, we first calculate the entropy for each column in the decision matrix and then use it to find the corresponding criterion weight. The criteria weights for the decision matrix formed by average QoS are given in Table 5. The criteria weights for each time slot of our experiment calculated using this method are given in Table 6.

Criteria	w_{c_1}	w_{c_2}	w_{c_3}	w_{c_4}
Weight	0.35	0.33	0.16	0.16

Table 5 Criteria weights calculated using the Entropy Method for decision matrix formed by average QoS (Table 4)

5.3 Results and discussion

Histograms of the CPU, memory and I/O for response time for services in the dataset are given in Figure-7. This shows that some of these measurements have a bi-modal frequency distribution or have a scattered distribution, which means that the mean (shown in Table 4) cannot effectively represent the entire data; thus in this scenario, MCDM based on average QoS (used in existing approaches) is not a reliable method for service selection.

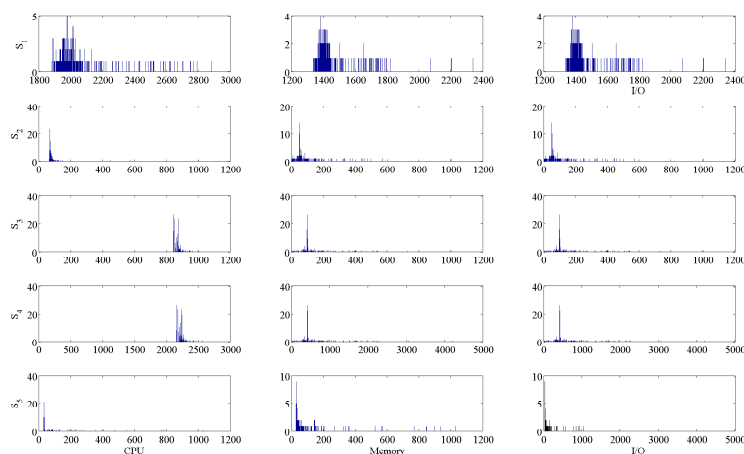


Fig. 7 Histograms of response times in the dataset

The final service selection results obtained using the three simulation models described in the previous sub-section are presented in Table 7. Service S_3 is selected by using simulation Model I, which uses the average of QoS values with TOPSIS. The same results are obtained by using ELECTRE in Model I.

Services	TOPSIS-Based Simulation Models				
	Model-I	Model-II	Model-III(a)	Model-III(b)	Model-III(c)
S_1	0.3646	0	0	0	0
S_2	0.9791	70	41.45526718	46.02430031	50.32104085
S_3	0.8183	33	20.2903275	24.12103461	25.14624907
S_4	0.3335	0	0	0	0
S_5	0.9733	197	99.99244968	121.5538862	146.2197723
Selected Service	S_2	S_5	S_5	S_5	S_5
Services	ELECTRE-Based Simulation Models				
	Model-I	Model-II	Model-III(a)	Model-III(b)	Model-III(c)
S_1	0	0	0	0	0
S_2	3	108	63.97683691	73.16884246	79.3544612
S_3	2	51	30.72870504	36.71901199	38.78382454
S_4	0	1	0.400004472	0.400054476	0.400663151
S_5	3	209	109.3630488	132.4192142	157.1707141
Selected Service	S_2	S_5	S_5	S_5	S_5

Table 7 Final service ranks with the simulation models

The results of the service selection (using TOPSIS and ELECTRE) using simulation Models II and III in each time slot are given in Figure-8, where S_5 has the highest rank in both models. Our proposed framework (Models II and III) leads to the selection of Service S_5 . Although, aggregation without time

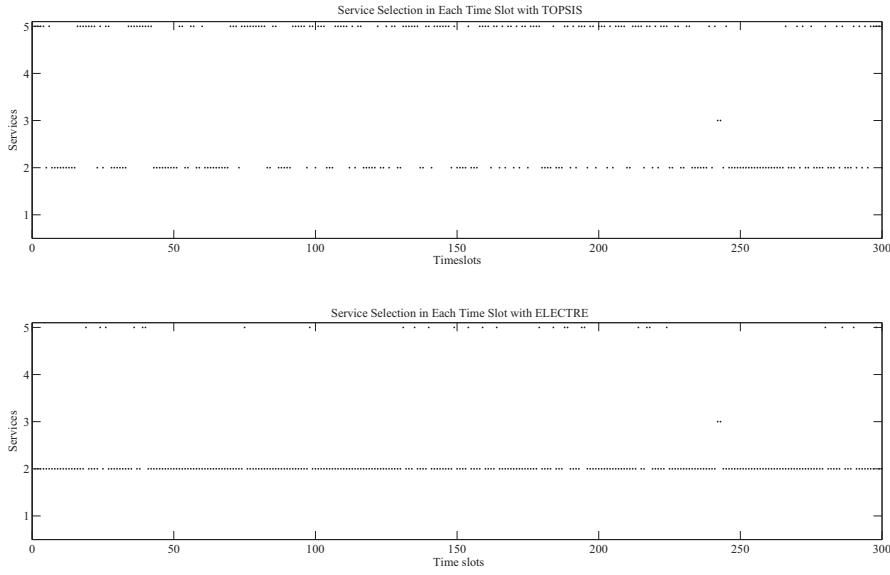


Fig. 8 Services selected in each time slot with fixed subjective criteria weights

slot weights in Model II and aggregation with variation in time slot weights in Model III leads to the selection of the same service, there is a considerable variation in the ranking values assigned by each model. This variation in rank values shows that having a weight for time slots is effective in controlling the relative importance of new and old QoS values. This is further evident from the difference in the aggregated output values calculated by using the three logistic decay curves to calculate the time slot weights (Models III(a), III(b) and III(c) in Table 7).

These results show that selecting a cloud service by using average QoS can lead to the selection of a service that has a better service average but is not the best service, due to the variations in QoS performance of IaaS cloud services. Our proposed approach is capable of taking these variations into account by considering the entire QoS history instead of using average QoS. This approach captures the variations in performance of services and gives more importance to recent QoS data without discarding the older QoS data (which is accorded less importance), which in turn leads to more reliable cloud service selection.

In simulation Model IV, the ability of our proposed approach to use different criteria weights in different time slots was assessed by using the entropy method [33] to dynamically calculate the criteria weights for each time slot. The final service selection results are given in Table 8 (wherein the superscript e denotes that the entropy weights have been used in the simulation models). Although the overall service ranks in simulation Model IV are the same as those obtained using fixed criteria weights (Table 7), there is nevertheless a variation in actual rank values assigned to each service, which suggests that in

Services	TOPSIS-Based Simulation Models				
	Model-I ^e	Model-II ^e	Model-III(a) ^e	Model-III(b) ^e	Model-III(c) ^e
S_1	0.4028	0	0	0	0
S_2	0.9771	34	19.53900812	21.31343187	22.94338047
S_3	0.831	2	1.561215687	1.985053963	1.999874971
S_4	0.3491	0	0	0	0
S_5	0.9913	264	140.6378205	168.4007352	196.7438068
Selected Service	S_2	S_5	S_5	S_5	S_5
Services	ELECTRE-Based Simulation Models				
	Model-I ^e	Model-II ^e	Model-III(a) ^e	Model-III(b) ^e	Model-III(c) ^e
S_1	0	0	0	0	0
S_2	3	56	35.16355974	39.06340062	40.96300738
S_3	2	3	2.328742316	2.976528361	2.999802544
S_4	0	0	0	0	0
S_5	4	272	146.2277299	174.6360333	203.3238442
Selected Service	S_2	S_5	S_5	S_5	S_5

Table 8 Final service ranks calculated in each simulation model with variable criteria weights

scenarios where users' criteria vary with time depending on changes in workload or predictable seasonal variations in business needs, our approach is able to use dynamic criteria weights to take these changes into account.

6 Conclusion and future work

In this paper, we discussed the cloud service selection problem and proposed a novel cloud service selection framework in which the QoS history is divided into several time slots. A service selection decision is taken at each time slot and all decisions are aggregated to find the overall optimal service. The decisions at time slot level are taken by applying TOPSIS or ELECTRE to the QoS data at each time slot along with the user criteria weights. We compared the results obtained using this approach with those obtained by existing approaches in which a MCDM technique is applied to average QoS data. We found that, due to the variations in service performance resulting from the dynamic nature of the cloud environment, the compared approaches do not lead to the selection of the same service. Furthermore, we found that the overall service rank also depends on the weights assigned to the time slots, which can be used to control the relative importance of older and newer QoS data in the decision making process. In addition to time slot weights, our proposed framework also permits the use of different criteria weights for each time slot. This feature is useful when there is a seasonal variation in service users' requirements, and as a result, the criteria weights also vary between time slots. The framework proposed in this paper deals with service selection in the pre-interaction period only. Work on post-interaction service migration decisions is needed, and several other important factors such as the cost of migration in terms of service disruption and data transfer, etc. also need to be included in the decision making process. Furthermore, there are several adjustable parameters in the

logistic decay function (Equation 18) and more work is needed to determine their optimal values for various decision making scenarios. This is our future work.

References

1. S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, A. Ghalsasi, Cloud Computing : The Business Perspective, *Decision Support Systems* 51 (1) (2011) 176–189.
2. S. K. Garg, S. Versteeg, R. Buyya, A framework for ranking of cloud computing services, *Future Generation Computer Systems* 29 (4) (2013) 1012–1023, ISSN 0167-739X.
3. Z. Zheng, X. Wu, Y. Zhang, M. Lyu, J. Wang, QoS Ranking Prediction for Cloud Services, *IEEE Transactions on Parallel and Distributed Systems* In Press, ISSN 1045-9219.
4. M. Behzadian, S. K. Otaghsara, M. Yazdani, J. Ignatius, A state-of-the-art survey of TOPSIS applications, *Expert Systems with Applications* 39 (17) (2012) 13051–13069, ISSN 0957-4174.
5. M. Pastaki Rad, A. Sajedi Badashian, G. Meydanipour, M. Ashurzad Delchah, M. Alipour, H. Afzali, A Survey of Cloud Platforms and Their Future, in: *Proceedings of the International Conference on Computational Science and Its Applications: Part I, ICCSA '09*, Springer-Verlag, Berlin, Heidelberg, ISBN 978-3-642-02453-5, 788–796, 2009.
6. J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, Q. Li, Comparison of Several Cloud Computing Platforms, in: *Second International Symposium on Information Science and Engineering (ISISE)*, 23–27, 2009.
7. R. Filepp, L. Shwartz, C. Ward, R. Kearney, K. Cheng, C. Young, Y. Ghosheh, Image selection as a service for cloud computing environments, in: *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, IEEE, 1–8, 2010.
8. A. Li, X. Yang, S. Kandula, M. Zhang, Comparing Public-Cloud Providers, *Internet Computing* 15 (2) (2011) 50–53, ISSN 1089-7801.
9. A. Li, X. Yang, S. Kandula, M. Zhang, Cloudcmp: Shopping for a cloud made easy, in: *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, USENIX Association, URL <http://research.microsoft.com/apps/pubs/default.aspx?id=136451>, 2010.
10. A. Li, X. Yang, S. Kandula, M. Zhang, CloudCmp: comparing public cloud providers, in: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, IMC '10*, ACM, New York, NY, USA, ISBN 978-1-4503-0483-2, 1–14, 2010.
11. A. Li, X. Zong, S. Kandula, X. Yang, M. Zhang, CloudProphet: towards application performance prediction in cloud, *SIGCOMM Comput. Commun. Rev.* 41 (4) (2011) 426–427, ISSN 0146-4833.

12. G. Nie, Q. She, D. Chen, Evaluation Index System of Cloud Service and the Purchase Decision- Making Process Based on AHP, in: L. Jiang (Ed.), Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011) November 19-20, 2011, Melbourne, Australia, vol. 112 of *Advances in Intelligent and Soft Computing*, Springer Berlin Heidelberg, 345–352, 2012.
13. J. Siegel, J. Perdue, Cloud Services Measures for Global Use: The Service Measurement Index (SMI), in: Annual SRII Global Conference (SRII), IEEE, 411–415, 2012.
14. S. Garg, S. Versteeg, R. Buyya, SMICloud: A Framework for Comparing and Ranking Cloud Services, in: Fourth IEEE International Conference on Utility and Cloud Computing (UCC), IEEE, 210–218, 2011.
15. S.-M. Han, M. M. Hassan, C.-W. Yoon, E.-N. Huh, Efficient service recommendation system for cloud computing market, in: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, ICIS '09, ACM, New York, NY, USA, 839–845, 2009.
16. J. Kang, K. M. Sim, Cloudle: A Multi-criteria Cloud Service Search Engine, in: IEEE Asia-Pacific Services Computing Conference (APSCC), 339–346, 2010.
17. J. Kang, K. M. Sim, Towards Agents and Ontology for Cloud Service Discovery, in: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 483–490, 2011.
18. J. Kang, K. M. Sim, Ontology and search engine for cloud computing system, in: International Conference on System Science and Engineering (ICSSE), 276–281, 2011.
19. C. Chen, S. Yan, G. Zhao, B. S. Lee, S. Singhal, A Systematic Framework Enabling Automatic Conflict Detection and Explanation in Cloud Service Selection for Enterprises, in: IEEE 5th International Conference on Cloud Computing (CLOUD), IEEE, 883–890, 2012.
20. S. Wang, Z. Liu, Q. Sun, H. Zou, F. Yang, Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing, *Journal of Intelligent Manufacturing* (2012) 1–9 ISSN 0956-5515.
21. W. Zeng, Y. Zhao, J. Zeng, Cloud service and service selection algorithm research, in: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, GEC '09, ACM, New York, NY, USA, 1045–1048, 2009.
22. M. Godse, S. Mulik, An approach for selecting software-as-a-service (SaaS) product, in: IEEE International Conference on Cloud Computing, IEEE Computer Society, 155–158, 2009.
23. Z. Rehman, O. K. Hussain, F. K. Hussain, S. Parvin, A Framework for User Feedback Based Cloud Service Monitoring, in: The Sixth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), IEEE Computer Society, Palermo, Italy, 257–262, 2012.
24. Z. Rehman, F. K. Hussain, O. K. Hussain, Towards Multi-criteria Cloud Service Selection, in: Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 44–48, 2011.

25. Umm-e-Habiba, S. Asghar, A survey on multi-criteria decision making approaches, in: International Conference on Emerging Technologies (ICET), IEEE, 321–325, 2009.
26. Y.-H. Hung, S.-C. T. Chou, G.-H. Tzeng, Knowledge management adoption and assessment for SMEs by a novel MCDM approach, *Decision Support Systems* 51 (2) (2011) 270 – 291.
27. T. Grbz, S. E. Alptekin, G. I. Alptekin, A hybrid MCDM methodology for ERP selection problem with interacting criteria, *Decision Support Systems* 54 (1) (2012) 206 – 214.
28. D. Petkov, O. Petkova, T. Andrew, T. Nepal, Mixing Multiple Criteria Decision Making with soft systems thinking techniques for decision support in complex situations, *Decision Support Systems* 43 (4) (2007) 1615 – 1629.
29. G. Kou, Y. Shi, S. Wang, Multiple criteria decision making and decision support systems, *Decision Support Systems* 51 (2) (2011) 247 – 249.
30. P. Tan, S. Lee, A. Goh, Multi-criteria decision techniques for context-aware B2B collaboration in supply chains, *Decision Support Systems* 52 (4) (2012) 779 – 789.
31. E. Triantaphyllou, B. Shu, S. Sanchez, T. Ray, Multi-criteria decision making: an operations research approach, *Encyclopedia of electrical and electronics engineering* 15 (1998) 175–186.
32. J. Lu, *Multi-Objective Group Decision Making: Methods Software and Applications with Fuzzy Set Techniques*, Series in Electrical and Computer Engineering, Imperial College Press, ISBN 9781860947933, 2007.
33. T.-C. Wang, H.-D. Lee, M.-S. Chang, A fuzzy TOPSIS Approach with entropy measure for decision-making problem, in: IEEE International Conference on Industrial Engineering and Engineering Management, IEEE, 124 –128, 2007.