CrossMark

RESEARCH PAPER

A Reinforcement Learning Based Model for Adaptive Service Quality Management in E-Commerce Websites

Hoda Ghavamipoor · S. Alireza Hashemi Golpayegani

Abstract Providing high-quality service to all users is a difficult and inefficient strategy for e-commerce providers, especially when Web servers experience overload conditions that cause increased response time and request rejections, leading to user frustration and reduced revenue. In an e-commerce system, customer Web sessions have differing values for service providers. These tend to: give preference to customer Web sessions that are likely to bring more profit by providing better service quality. This paper proposes a reinforcement-learning based adaptive e-commerce system model that adapts the service quality level for different Web sessions within the customer's navigation in order to maximize total profit. The e-commerce system is considered as an electronic supply chain which includes a network of basic e- providers used to supply e-commerce services for end customers. The learner agent noted as e-commerce supply chain manager (ECSCM) agent allocates a service quality level to the customer's request based on his/her navigation pattern in the e-commerce Website and selects an optimized combination of service providers to respond to the customer's request. To evaluate the proposed model, a multi agent framework composed of three agent types, the ECSCM agent, customer agent (buyer/browser) and service provider agent, is employed. Experimental results show that the proposed model improves total profits through cost

reduction and revenue enhancement simultaneously and encourages customers to purchase from the Website through service quality adaptation.

1 Introduction

E-commerce is one of the sophisticated services that is based on a significant part of the basic e-services offered on the Internet, such as data storage, networking, e-payment, and search. The advent of service-oriented architecture and distributed services, such as cloud services, enable e-commerce service providers to supply their basic e-services using external providers.

E-commerce service, like other sophisticated services, can be considered a supply chain of basic electronic services. Suppliers in this structure can provide different services with a variety of Quality of Service (QoS) levels and prices that are available on demand. This supply chain approach facilitates QoS adaptation and personalization of e-commerce service delivery.

QoS adaptation originates from the concept that uniform service quality allocation for all users is not an efficient strategy, especially for heavily loaded service providers, such as e-commerce Websites. There are different classes of Web sessions that have different values for service providers. A Web session is a sequence of related requests that together achieves a higher level user goal. In an e-commerce system, Web sessions that are likely to lead to a purchase and bring more profit are more valuable. Consequently, e-commerce service providers try to retain these

H. Ghavamipoor · S. A. Hashemi Golpayegani (✉)
Computer and Information Technology Engineering Department, Amirkabir University (Tehran Polytechnic), Tehran, Iran
e-mail: sa.hashemi@aut.ac.ir

H. Ghavamipoor
e-mail: h.ghavamipoor@aut.ac.ir

Springer

valuable customers by satisfying them through better service quality in order to gain the highest possible profit from their business. For this purpose, some research in the area of information systems and especially e-commerce systems has been undertaken to realize differentiation in offering e-services to different Web sessions (Bhatti and Friedrich 1999; Li and Jamin 2000; Urgaonkar and Shenoy 2005; Urgaonkar 2005; Yue and Wang 2007; Ewing and Menascé 2009; Totok and Karamcheti 2010; Suchacka and Borzemski 2013; Lakshmi et al. 2017; Poggi 2014). One of the common methods of service differentiation and admission control in the literature is prioritization (Totok and Karamcheti 2010; Poggi 2014; Lakshmi et al. 2017). Using this method, more valuable users get higher priority for receiving services and using the system's resources. Therefore, the likelihood that valuable users will complete their purchasing process and generate revenue for the system increases as they receive better service quality. To identify the value of each user, there are both static and dynamic approaches. In the static approach (Bhatti and Friedrich 1999; Li and Jamin 2000), users are classified in different categories according to their history, and the predetermined class for each user during his/her navigation on the Website does not change. Contrary to this, in the dynamic approach (Totok and Karamcheti 2010; Poggi 2014) the class of each customer is detected based on his/her behavior in the current session, so it considers the changes in the user behavior throughout his/her navigation.

The contribution of this paper is to propose an RL-based e-commerce supply chain management model that adapts the service quality level for different Web sessions within the customer's navigation in order to maximize total profit. In the proposed model the adaption method is used instead of prioritization as in the literature. This method is more flexible because the different customer classes in prioritization are predefined and each customer should be assigned to one of them, but in adaptation, according to the situation of the system including its traffic load, the number of different levels of customers and other related factors, different e-service qualities are presented to different users.

Furthermore, the proposed model is developed at the e-commerce supply chain level (it is not limited to an e-commerce Website). The model selects a QoS-level and basic e-service providers to fulfill requests based on the customer's navigational pattern. This adaptation strategy identifies the customers with purchase oriented navigational patterns and supports them by providing service with higher quality levels while it decreases the service quality level of customers who browse the Webpages over and over again without buying anything. The proposed model is expected to improve total profit through cost reduction and revenue enhancement simultaneously.

The organization of the paper is as follows. In Sect. 2, the related papers from two main areas are reviewed: (1) service differentiation and prioritization, and (2) e-commerce systems adaptation. In Sect. 3, some concepts and techniques utilized in the paper are briefly explained. The methodology of obtaining the proposed model is presented in Sect. 4. The experimental setup and evaluation results are provided in Sect. 5 to show the effectiveness of the proposed model. Section 6 contains conclusions and suggests some future work.

2 Related Work

In this section, we describe related work from two main perspectives: (1) admission control and service differentiation that prioritize classes of customers, and (2) adaptive e-commerce systems that automatically adapt to different customer classes.

One of the earliest papers is from Bhatti and Friedrich (1999) who developed the WebQoS mechanism to control QoS in e-commerce Web servers. It classifies requests into two distinct classes: basic and premium. Basic requests can be rejected to maintain the QoS of premium requests in order to preserve server performance. The WebQoS does not have any customer class detection mechanism. Later, Li and Jamin (2000) designed an algorithm to distribute the network bandwidth and the processing capacity of a Web server among several classes of consumers. In this approach, the assignment of resources to customers is predefined, and requests from a customer class that does not have available resources are rejected despite the existence of resources for other customer classes.

Urgaonkar and Shenoy (2005) proposed an admission control mechanism specifically designed to remain operational under extreme overloads. It increases its efficiency by sorting requests into classes and admitting or rejecting sets of requests instead of individual requests. Later, Urgaonkar (2005) introduced the session concept into the mechanism. Yue and Wang (2007) developed a session based admission control system for e-commerce servers by classifying the clients into two classes: premium if they have purchased products previously, and basic if they have not. Under overload conditions, the system first rejects the basic clients.

A business-oriented approach was presented by Ewing and Menascé (2009) to dispatch incoming requests to servers and allocate servers to server clusters according to customer priority classes. They proposed a model for self-optimization through request prioritization and self-configuration by the dynamic assignment of servers to server clusters. All of the above studies classify customers based on previous navigation patterns and assume that current

behavior is similar to previous behavior. They do not consider the current dynamic interaction of users.

Totok and Karamcheti (2010) developed a mechanism that dynamically assigns higher priority to requests that have the potential to provide a higher reward, typically higher revenue. They considered the navigational pattern of customers to prioritize customers, but their work does not have any self-configured resource management mechanism to respond to customer requests. Suchacka and Borzemski (2013) proposed a method for priority-based admission control at the Web server system to differentiate Quality of Service (QoS) with regard to user-perceived delays. To detect and cope with the system overload, a kind of a load indicator was proposed, based on online measurements of page response times. They showed that their admission control mechanism is capable of providing key customers with limited delays while improving QoS for ordinary customers under heavy load.

Simultaneous with the extensive commercial use of cloud computing systems, some admission control systems in this context were proposed (Lakshmi et al. 2017; Poggi 2014). Despite the previous studies, these models configured the required cloud infrastructure (including different external resources) to present different QoS levels for predefined classes of sessions. Consequently, choosing an optimal combination of cloud resources to present the final e-service was an issue in these works.

Lakshmi et al. (2017) proposed an admission control and resource allocation mechanism in a cloud computing context. However, although this cloud-based mechanism expanded the previous models which were applicable in the limited context of an e-commerce Website, it cannot be utilized in the e-services supply chain layer. This is due to the fact that in an IaaS cloud computing context, there are a variety of resources managed by a cloud provider. Indeed, cloud resources are not autonomous and independent providers. But our model proposes a dynamic mechanism for assigning user service requests to autonomous external service providers which have their independent service providing strategies and service conditions including the type of services, the provided service quality levels, pricing strategy etc.

Furthermore, Lakshmi et al. (2017) utilized Support Vector Machine (SVM) and Artificial Neural Network (ANN) to detect the classes of customers statically and allocate the appropriate resources to each request. Both of these methods belong to the category of supervised machine learning which relies on the fact that there is prior knowledge of what the output values for an input sample should be. Therefore, the goal of these methods is to produce a function that, given a sample of data and desired outputs, best approximates the relationship between input and output observable in the data. But in the context of our problem, the detection of the customer classes is done dynamically throughout their navigation in the Website and the requests are allocated to a supply chain layer which includes a variety of autonomous providers. Therefore, due to the complexity and high dynamics of the context, correct input/output pairs cannot be provided for training the model. To tackle the issue, the proposed model uses reinforcement learning. This method is a goal-oriented learning based on interaction with the environment. Reinforcement learning allows the agent to learn their behavior based on feedback from the environment without need of prior knowledge.

The adaptation mechanism proposed in this paper differs from previous related work in two ways. Firstly, it presents an online adaptive QoS allocation to customers according to their dynamic QoS-sensitive navigational behavior in e-commerce Websites. This adaptation is updated and improved over time using reinforcement learning techniques. Secondly, it proposes a dynamic mechanism for assigning user service requests to autonomous external service providers that have properties which do not resemble the assignment of user service requests to internal Web servers. Indeed, the proposed model applies an adaptation mechanism on the e-service supply chain layer.

The differences and similarities of our work in relation to previous studies regarding admission control and service differentiation are presented in Table 1.

From the second viewpoint (i.e., adaptive e-commerce systems), various adaptation features are considered in the literature to enable e-commerce systems to provide customer-centric services. Brusilovsky et al. (2007) classified the adaptation features into three main categories:

Content adaptation suggests products/services or configuration styles for complex products/services tailored to the users. This adaptive system actively guides the customer by displaying the proper contents to satisfy his/her particular needs and preferences (Sarwar et al. 2001; Aciar et al. 2007; Schafer et al. 2007; Chan et al. 2012; Li et al. 2013; Chen et al. 2015).

Presentation adaptation tailors the presentation of contents to user needs considering context, device, or location. For instance, the presentation may use different media, such as written text, speech, and pictures; it may adopt different presentation styles (verbose, synthetic, more or less detailed, simple or technical). The system may also personalize the kind of information about products and services it presents according to the context of the user (Hong et al. 2009; Rosaci and Sarné 2012; Belk et al. 2015).

Structure adaptation personalizes the link structure between pages to support customer navigation. Different techniques, such as the adaptive sorting of links (displaying a list of links from best to worst), adaptive hiding of

Table 1 Review of the related works in admission control and service differentiation area

References	Request differentiation policy	Self-configuration of resources?	Session based method?	Dynamic classification of customers?	Optimization goal	Configuration layer
Aciar et al. (2007)	Prioritization	Yes	No	No	Maximization of throughput	Resource layer
Al-Masri and Mahmoud (2007)	Prioritization	Yes	No	No	Maximization of bandwidth utilization	Resource layer
Belk et al. (2015)	Prioritization	Yes	Yes	No	Maximization of revenue	Resource layer
Bellifemine et al. (2001)	Prioritization	No	Yes	No	Maximization of revenue	Resource layer
Bhatti and Friedrich (1999)	Prioritization	Yes	Yes	No	Maximization of revenue	Resource layer
Brusilovsky et al. (2007)	Prioritization	No	Yes	Yes	Maximization of profit	Resource layer
Chan et al. (2012)	Prioritization	No	Yes	No	Maximization of throughput	Resource layer
Chen et al. (2015)	Prioritization	Yes	Yes	No	Minimization of costs	Cloud resources layer
Our work	Adaptation	Yes	Yes	Yes	Maximization of profit	Supply chain layer

links (hiding inappropriate links), adaptive annotation of links (marking links as appropriate or inappropriate but not deleting any) and map adaptation (changing graphical overviews of the link structure), are utilized. One of the main goals of adaptation in this area is to guide the customers to navigate the purchase pattern (Bathumalai 2008; Raufi et al. 2010; Chen and Ryu 2013; Yin and Guo 2013).

Our work adds one more adaptation dimension, service quality adaptation tailored to users considering their navigation patterns. The proposed adaptation strategy identifies the customers with purchase navigational patterns and encourages them to complete the purchase process by providing services with higher quality levels, and it decreases the service quality level of customers who browse the Webpages again and again without buying anything. The proposed model improves total profit through cost reduction and revenue enhancement simultaneously. Some of the previous work dealing with different adaptation dimensions is depicted in Table 2.

The focus of adaptive e-commerce systems in the literature appears to be on improving system performance even though the e-commerce service providers are seeking higher profits from their business. The goal of our adaption is to increase total profit, and we try to obtain this goal through adaptive QoS allocation and dynamic supply chain configuration.

3 Related Concepts and Techniques

3.1 Electronic Services Supply Chain

Electronic services supply chain (Ghavamipoor and Golpayegani 2013) includes a network of basic e-service suppliers and network service providers that are employed by a final service provider to provide services for end customers. A sample structure of e-commerce is described below (Fig. 1).

In this structure, basic e-service providers – called e-service suppliers – provide search, e-mail, storage and e-payment services for the e-commerce service provider. The e-commerce service provider – called e-service producer – assembles the basic e-services and prepares the final service. The e-service producer presents the final composite service to the customer directly or through some other Websites called e-service distributors. Between each tier of the supply chain, there are network providers to interconnect providers of different levels.

3.2 Quality of Service (QoS) in E-Commerce

In e-commerce, QoS is defined as one of the e-commerce triple quality dimensions: (1) Quality of system, e.g., Website design, access convenience, ease of use, (2) Quality of information, e.g., accuracy, usefulness and

Table 2 Review of the related work in adaptive system area

| References	Adaptation feature	Utilized technique	Goal of adaptation
Ghavamipoor and Golpayegani (2013)	Content adaptation	Collaborative filtering	Recommendation effectiveness
Ghavamipoor and Golpayegani (2016)	Content adaptation	Text mining and ontology's Information structure	Recommendation effectiveness
Ghavamipoor and Golpayegani (2017)	Content adaptation	Collaborative filtering	Recommendation effectiveness
Larisa et al. (2014)	Presentation adaptation	Agent technology and preference and association rules	Recommendation effectiveness
Lee and Kozar (2006)	Presentation adaptation	Agent technology	Recommendation effectiveness
Li and Jamin (2000)	Presentation adaptation	Classification and adaptation rules	Improving e-commerce system's performance
Li et al. (2013)	Structure adaptation	Clustering and web usage mining	Improving website's organization
Lin (2007)	Structure adaptation	Link matrix and similarity matrix	Improving system's performance
Mark and Csaba (2007)	Structure adaptation	Mathematical programming	More convenient user navigation
Menascé (2002)	Structure adaptation	Tabu search and web usage mining	Improving e-commerce system's performance
Our work	QoS adaptation	Reinforcement learning	Profit maximization
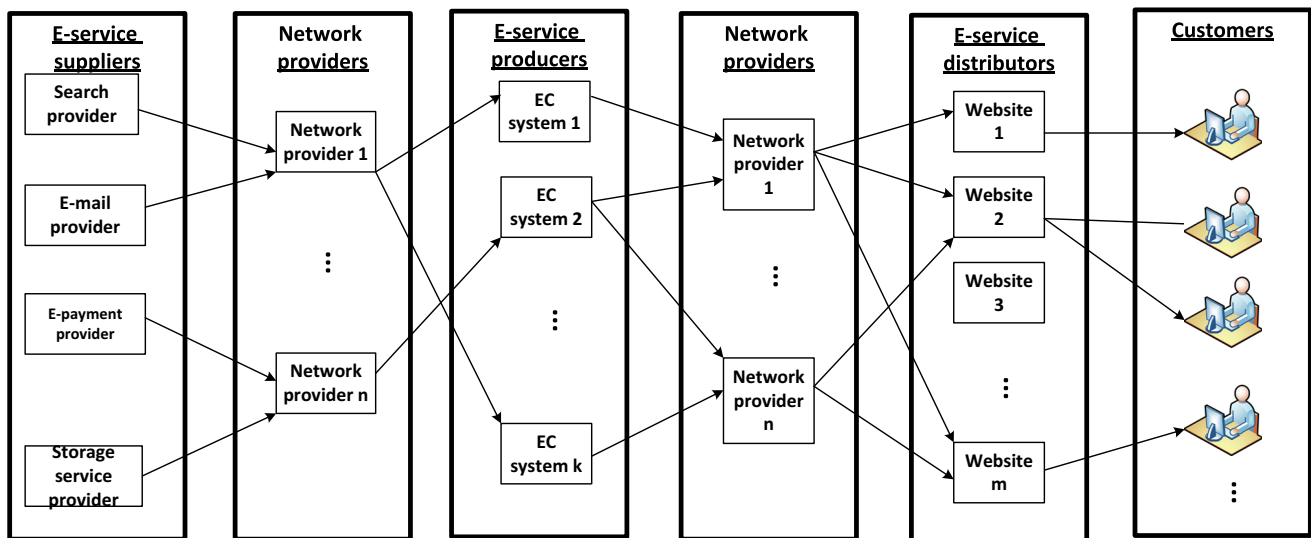
Fig. 1 A sample structure of an e-commerce service supply chain

completeness of data, and (3) Quality of service (Lee and Kozar 2006; Lin 2007).

QoS is widely employed to describe the nonfunctional characteristics of Web services. Several studies have attempted to enumerate the different dimensions of e-service quality. For instance, response time, throughput and failure probability are e-service quality dimensions, employed by Zheng et al. (2014). Menascé (2002) also included availability, security, response time, and the throughput as e-service quality aspects. Among these service quality dimensions, response time is more likely to be measurable than other dimensions of modeling customer behavior. Furthermore, in the context of service differentiation and adaptation in the literature, response time is

sometimes referred to as the QoS provided to users (Totok and Karamcheti 2010; Suchacka and Borzemski 2013; Harini and Padmanabhan 2013; Larisa et al. 2014; Zatwarnicki and Zatwarnicka 2014). Therefore, in this paper, response time is utilized as the QoS measure, which is defined as the time taken to send a request and receive a response (Al-Masri and Mahmoud 2007). In the remainder of this paper, we assume that there are excellent, average and poor service quality levels, which are noted as R1, R2 and R3, respectively.

3.3 QoS Sensitive Customer Behavior Model (QoS-CBMG)

QoS-CBMG (Ghavamipoor and Golpayegani 2017) is a directed graph with transition probabilities, with each node representing a Webpage and edges standing for customer navigation through the Webpages. A typical session in this model consists of the following requests: a user starts Website navigation by accessing the entry page, searches for particular products (Search), visits the product catalogs (Browse), selects specific items (Select), adds some of them to the shopping cart and initiates the check-out process (Add To Cart), and finally commits the order and pays the total price (Pay). In each state (except Entry), the customer may abandon his/her navigation and exit the Website.

Each QoS-CBMG represents the behavior of a class of users such as browsers or buyers. In the following, two QoS-CBMGs for a sample e-commerce Website are illustrated: QoS-CBMG for Buyers (Fig. 2a) and QoS-CBMG for Browsers (Fig. 2b). In QoS-CBMGs, edge labels show the probability that a customer navigates from a Webpage to another based on perceived service quality. For example, assuming the user is at browse state, he/she will navigate to search state with a probability of 0.3 for all three considered service quality levels (R1, R2 and R3) if the user is a buyer (Fig. 2a).

3.4 Reinforcement Learning (RL)

Reinforcement learning (Sutton and Barto 1998) is a field of study in machine learning where an agent, by interacting with and receiving feedback from its environment, attempts to learn an optimal action selection policy. An RL model generally has 4 basic components:

- A set of environment states S;
- A set of actions A;
- Rules of transitioning between states;
- Rules that determine the scalar immediate reward of a transition.

In an RL model, an agent is connected to its environment via perception and action, as depicted. At each step of interaction, the agent receives the current state, s, of the environment; the agent then chooses an action, a, to generate as output. The action changes the state of the environment, and the value of this state transition is communicated to the agent through an immediate reward, r. The agent's behavior should choose actions that tend to increase the long-run sum of values of the reward.

3.5 Non-deterministic Q-Learning

A commonly used RL algorithm is Q-Learning (Sutton and Barto 1998), a model-free RL algorithm that assumes that the agent has no explicit knowledge of the environment's behavior prior to interacting with it. The goal of this algorithm is to learn the state-action pair value, Q(s, a) that represents the long-term expected reward for each pair of state and action. Due to the nature of the problem, in this paper the modified version of the Q-learning algorithm that is suitable for a non-deterministic environment is used. In this algorithm, the action transition functions have probabilistic outcomes. Therefore, the reward function $r(s, a)$ and the transition function $\delta(s, a)$ are considered as probability distributions over outcomes based on s and a. The procedure of the non-deterministic Q-learning algorithm is as follows:

1. Initialize the $\tilde{Q}(s, a)$ value functions arbitrarily
2. Perceive the current state, s_0
3. Following a certain policy (e.g., e-greedy), select an appropriate action (a) for the given state (s_0)
4. Execute the selected action (a), receive immediate reward (r), and perceive the next state s_1
5. Update the value function as follows:

$$\tilde{Q}_n(s,a) \leftarrow (1 - \alpha_n)\tilde{Q}_{n-1}(s,a) + \alpha_n\left[r(s,a) + \gamma \max_{a'} \tilde{Q}_{n-1}(s',a')\right]$$

where

$$\alpha_n = \frac{1}{1 + visits_n(s,a)} \tag{1}$$

6. Let $s_0 = s_1$
7. Go to step 3 until state s_0 represents a terminal state
8. Repeat steps 2–7 for a number of episodes.

Each iteration of steps 2–7 represents a learning cycle, also called an "episode". In Eq. (1), s and a are the state and action updated during the nth iteration, $visits_n(s, a)$ is the total number of times this state-action pair has been visited up to and including the nth iteration, α_n is the learning rate parameter, and γ is the discount rate and impacts the present value of future rewards.

Fig. 2 a QoS-CBMG for buyers (Larisa et al. 2014). **b** QoS-CBMG for browsers (Larisa et al. 2014)
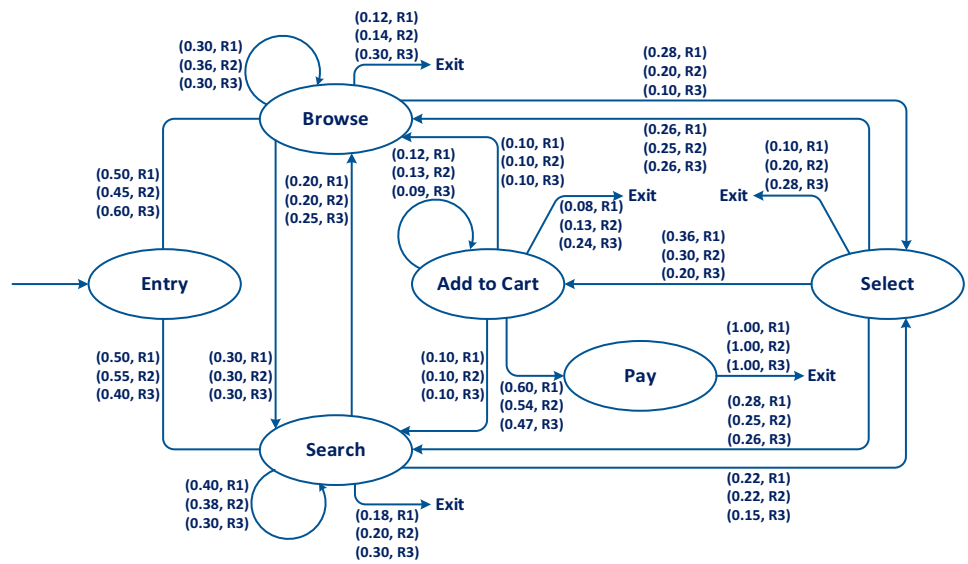
(a)
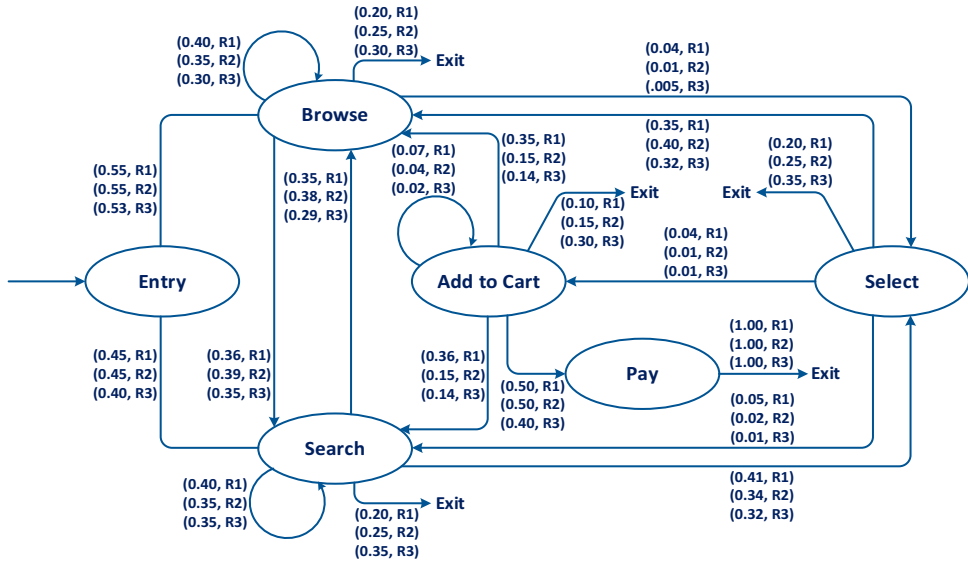
(b)

4 The RL-Based Methodology

To generate the proposed adaptive model, RL is employed as a means of QoS adaptation. In this framework, the learner agent is e-commerce supply chain manager (ECSCM) agent that selects the QoS level and basic e-service providers to respond to each customer's request. The main contribution of the paper is to propose an adaptive decision model for the ECSCM agent in order to maximize the e-commerce total profit. The conceptual model including the ECSCM agent and its interactions in an e-commerce supply chain is illustrated in Fig. 3.

As shown in Fig. 3, in each step of user navigation in the e-commerce Website, the customer (buyer or browser) clicks on the e-commerce Website's URLs and sends a service request to the ECSCM agent. According to the navigational pattern of the customer in the current session, the ECSCM agent selects the appropriate QoS level and basic e-service providers to supply the requested service. It constructs the best e-commerce service chain for the user request or user session. In the selected combination of eservice providers, each provider presents e-services in one or more levels (e.g., supplier, producer, network provider, distributor layers) of the constructed supply chain. The ECSCM agent forwards the requests to the selected providers. Then, provider agents supply the basic e-services and charge the e-service cost to the ECSCM agent. The providers notify the ECSCM agent if their supply capacity
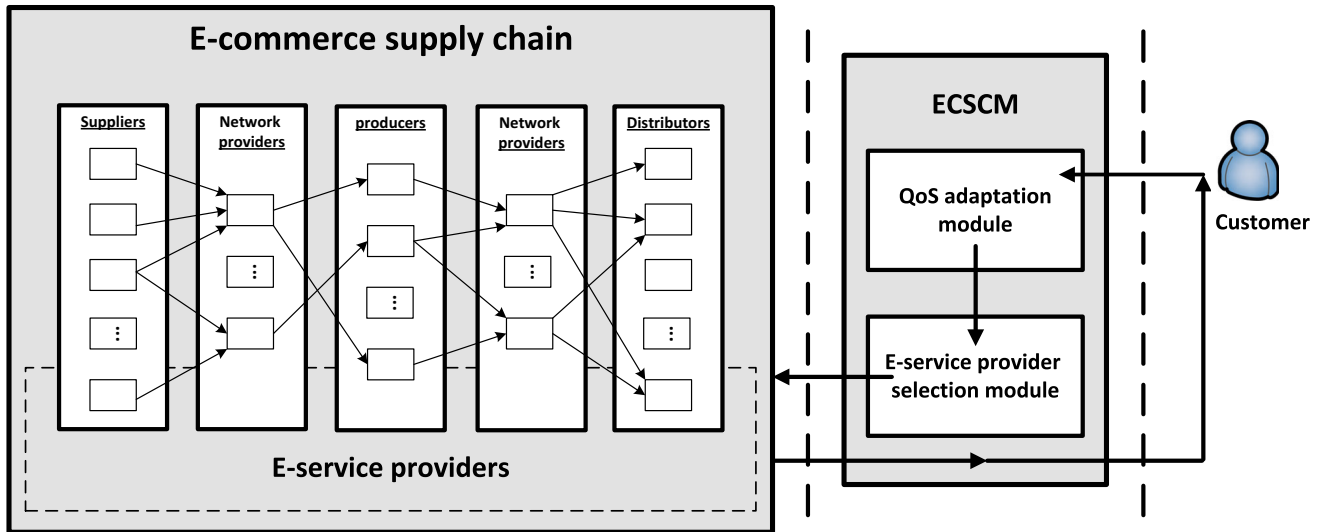
Fig. 3 The proposed multi agent framework

is full and they cannot provide their services or if their full capacity has been released and they can provide their services.

4.1 Formal Modeling

Before a detailed description of the methodology, the notations and the model's formal expression used throughout the rest of the paper are defined as follows.

    Sets

- Customers: C = {1,..., c}
- Basic e-service providers: Prov = {1, ..., p}
- Basic e-services: S = {1, ..., s}
- QoS levels of basic e-services: QoS = {1, ..., q}
- E-commerce Website's URLs: Pg = {1, ..., k}
- Presented QoS levels for customer to respond the requested URL: CQoS = {1, ..., w}
- Clickstream of customer c in his/her current session: $\{n_{1c}, \ldots, n_{lc}\}$
- Target URLs: Pgbuy = {1,..., u}

Parameters

- $SUPCost_{sqp}$: Cost of providing s_s with QoS_q by $Prov_p$
- $SUPCap_{sqp}$: Capacity of providing s_s with QoS_q by $Prov_p$
- $SUPQ_{sqp}$: The number of current assigned requests to $Prov_p$ for providing s_s with QoS_q
- m_c: Number of requests of c_c in his/her current session
- AoP_c: Purchase amount of c_c in the current session
- Ra_k: Rank of Pg_k
- Rep_k: The significance of repetition of Pg_k = {1 if Repetition is important, 0 Otherwise}

- $NRep_{kc}$: Number of Pg_k repetitions in the current session of c_c
- DEF: A default value
- Y_{sqp}: Providing of s_s with QoS_q by $Prov_p = \{0 \ if \ not \ provided, \ 1 \ if \ provided\}$
- $ReqPg_k = \{s_i, \ldots, s_j\}$: Required basic e-services to present Pg_k

Variables

- $SCQoS_{wkc}$: Selected $CQoS_w$ for providing Pg_k for c_c
- $SProv_{wk}$: A subset of Prov where $X_{pkw} = 1$
- $X_{pkw} = \{1 \ if \ Prov_p \ is \ selected \ to \ supply \ Pg_k$ $with \ CQoS_w, \quad 0 \ otherwise\}$

- Selected subset of available Prov to supply $SCQoS_{wkc}$ where:

$$\{\{\forall s_s \in S | (Pg_k, s_s) = 1\} \quad \exists! \ prov_p \in SProv_{wk} | (prov_p, s_s) = 1\}$$
$$and \ \{\nexists \ prov_p \in SProv_{wk} | (prov_p, s_s) = 0 \ and \ (URL_k, s_s) = 1\} \ and$$
$$Min\left(\sum_{p \subset SProv_{wk}} \sum_s SUPCost_{sqp} \cdot R1(pg_k, s_s) \cdot R2\left(P_p, s_s\right)\right)$$

  Goal function

-

$$MAX\left(\sum_c AoP_c - \sum_c \sum_{l=1}^{m_c} cost \ of \ providing \ services\right) \quad (2)$$

4.2 ECSCM Modeling

In the remainder of this section, we focus on the ECSCM modeling methodology, including QoS level allocation and

provider selection, to maximize the e-commerce total profit.

The ECSCM modeling methodology utilizes RL technique to train the ECSCM agent for QoS adaptation and linear programming optimization technique for provider selection. Three main steps to obtain the proposed model includes: (1) QoS adaptation, (2) provider selection and (3) adaptive supply chain management.

4.2.1 QoS Adaptation

To enable the ECSCM agent to adapt the QoS level to the customer navigational pattern, RL method is used. RL method provides a way to solve complex, real-world adaptation problems. As opposed to the supervised learning approaches whereby an agent learns from examples provided by a knowledgeable external supervisor (Weiss 1999), RL requires that the agent learn by directly interacting with its environment and responding to the receipt of rewards or penalties based on the impact each action has on the environment. In our problem, the ECSCM agent is considered as an RL learner agent and customers are considered as the environment, as well. As shown in Fig. 4, the ECSCM agent learns how to adapt QoS levels to customer requests to gain the maximum reward, i.e., the highest profit. Four basic components of RL (noted in Sect. 4) are presented as follows.

4.2.2 A Set of Environment States

The states of the customers are the states of the QoS-CBMG: Entry, Browse, Search, Select, Add to cart, Pay and Exit (described in Sect. 3.3). At each episode of the RL algorithm, each customer in his\her navigation is placed in one of these states. The environment state is composed of the individual states of all customers.

4.2.3 A Set of Actions

The ECSCM agent in response to each customer request to navigate to a new state has three available actions: to represent the service for the customer with service quality levels: (1) R1 or (2) R2 or (3) R3.
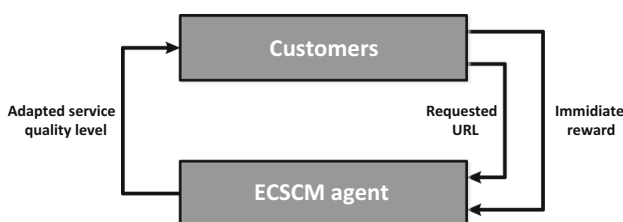

Fig. 4 The considered reinforcement learning model

4.2.4 Rules for Transitioning Between States

The customers as the components of the environment are transitioning between states. The behavior of customer in the simulation section (Sect. 5) is modeled according to the QoS-CBMG. If a customer is a buyer, then he/she navigates the e-commerce Website based on the transition probabilities of the QoS-CBMG for buyers considering his/her perceived service quality level and if a customer is a browser, then he/she navigates the e-commerce Website based on the transition probabilities of the QoS-CBMG for browsers considering his/her perceived service quality level.

4.2.5 Reward Function

A reward is provided to the ECSCM agent after executing a given action. Since the final goal of the proposed model is to maximize the total profit of the Website, the immediate reward function is defined as an expression of the potential profit of each transition. The definition of this function is presented as follows:

At first, we define the value function for each state of customer navigation (Eq. 3). This formula is composed of the rank of each URL $n_{lc}(Ra_{n_{lc}})$ and a coefficient of the rank of the number of that page's repetitions $(Ra_{n_{lc}} \times (NRep_{n_{lc}c} - 1))$ in the current session. $Ra_{n_{lc}}$ is a measure of proximity of URL_{nl_c} to purchase and the repetition of URL_{nl_c} in the current session increases this proximity and consequently increases the customer state value.

$$\text{Val}_{n_{lc}} = \text{Ra}_{n_{lc}} + \beta \, \text{Rep}_k\big(\text{Ra}_{n_{lc}} \times \big(\text{NRep}_{n_{lc}c} - 1\big)\big) \quad (3)$$

Based on the value of each customer state, the potential income of the state is calculated as follows (Eq. 4):

$$\text{I}_{nl_cc} = \begin{cases} \text{AoP}_c, & \text{If} \quad \text{Pg}_{nl_c} \in \text{Pgbuy} \\ \text{DEF} \cdot \text{Val}_{n_{lc}}, & \text{If} \quad \text{Pg}_{nl_c} \notin \text{Pgbuy} \end{cases} \quad (4)$$

In Eq. 4, if the state is a purchase page, then the income is equal to the purchase amount (AoP_c). Otherwise, the potential income is a coefficient of the state's value.

Finally, the immediate reward function for each state of a customer is defined as the potential profit of each state that is calculated as the total cost of the customer in his/her current session (up to this state) minus the potential income of this state (5).

$$\text{Reward} = I_{nl_cc} - \sum_{l}^{m_c} Cost_{URL_{n_{lc}}, CQoS_{n_{lc}}} \quad (5)$$

4.2.5.1 Training the Model and Creating Q-Table The most important and popular model-free RL algorithm, Q-Learning (Sect. 3), is utilized to quantify the preference and effectiveness of selecting an action given a perceived

state. Q-learning can handle problems with stochastic transitions and rewards. It has been proven that for any finite MDP, Q-learning eventually finds an optimal policy (Sutton and Barto 1998), in the sense that the expected value of the total reward return over all successive steps, starting from the current state, is the maximum achievable. Mark and Csaba (2007) introduced a method for transforming CBMG graphs to non-deterministic Markov chains. Since QoS-CBMG is an extended version of CBMG where the number of edges is multiplied by the levels of service quality, it assured that the environment in this problem is a non-deterministic MDP and this confines that Q-learning algorithm can be utilized in this context.

Following every selection of an action, the corresponding Q value is updated as Eq. 1 in order to construct a Q-table. As noted in Sect. 4.2.2, there are 7 different states for the environment and also according to Sects. 4.2.3, 3 possible actions are allowed for ECSCM in each state. Therefore, the Q-table is a $(7 * 3) \times (7 * 3)$ matrix. Each element of the Q-table indicates the quality of the ECSCM's action in each environment state. The Q-table is used by ECSCM to answer the current request of the customer according to his/her current navigational pattern. The interactions between ECSCM and the customers in each episode of the Q-learning algorithm are as follows (Fig. 4):

- *Current environment state* Customer sends his/her request for a URL.
- *ECSCM action* ECSCM chooses the service quality level for responding to the request.
- *Environment transition* Customer according to perceived service quality level, sends a new request.
- *Reward achievement* ECSCM receives the reward of his action according to the customer's transition.

The episodes are repeated until the learning process of the ECSCM agent is completed.

4.2.6 Service Provider Selection

To provide the requested URL for the customer, the first step of the methodology determines the appropriate QoS level. In the second step, the ECSCM agent selects the optimized (minimum cost) combination of the basic e-service providers to supply the determined QoS level.

It is assumed that there are multiple basic e-service providers who provide one or more basic e-services with multiple criteria (QoS levels and prices) and capacity constraints. ECSCM agent generates and maintains an L-table like Table 3 which determines the optimized (minimum cost) combination of the basic e-service providers to present each URL with a determined QoS level. L-table is updated if one of the following events occurs:

Table 3 A sample L-table

Requested URL	s_1	s_2	…	s_s
URL_1, $CQoS_1$	$Prov_1, QoS_1$	0	…	$Prov_{10}, QoS_2$
URL_1, $CQoS_2$	$Prov_3, QoS_2$	$Prov_1, QoS_1$	…	$Prov_7, QoS_3$
URL_1, $CQoS_3$	$Prov_2, QoS_2$	$Prov_8, QoS_3$	…	$Prov_1, QoS_3$
URL_2, $CQoS_1$	$Prov_9, QoS_1$	0	…	0
…	…	…	…	…

- One of the e-service providers of the combination reaches full capacity.
- One of the e-service providers of the combination becomes down.
- One of the full capacity e-service providers responds to its requests and becomes available.
- One of the down e-service providers becomes available.

For generating each row of the L-table and selecting the optimized combination of basic e-service providers, the following decision model is used. This model utilizes mixed integer linear programming (MIP) which is one of the common technics for supplier selection in supply chain management literature (Sutton and Barto 1998).

$$Min\left(\sum_{s \in ReqPg_k} \sum_{p=1}^{P} SUPCost_{sqp} \cdot X_{psq} \right) \qquad (6)$$

Subject to :

$$\sum_{s \in ReqPg_k} X_{psq} = 1 \quad \forall q \in QoS \text{ and } \forall p \in Prov \qquad (7)$$

$$CQoS_w \leq \sum_{s \in ReqPg_k} \sum_{p=1}^{P} QoS_q \cdot X_{psq} < CQoS_{w+1}, \quad \forall q \in QoS \qquad (8)$$

$$Y_{sqp} \cdot X_{psq} = 1, \quad \forall q \in QoS \text{ and } \quad \forall p \in Prov \text{ and } \forall s \in ReqPg_k \qquad (9)$$

$$SUPQ_{sqp} + 1 \leq, \quad \forall q \in QoS \text{ and } \quad \forall p \in Prov \text{ and } \forall s \in ReqPg_k \qquad (10)$$

$$X_{psq} \in \{0, 1\} \qquad (11)$$

In this model, the objective function (6) presents the minimization of total service cost for providing Pg_k with $CQoS_{w+1}$. Constraint (7) confirms that each basic e-service is assigned to exactly one supplier. Constraint function (8) ensures that the sum of response time of basic e-services preserves the desired service quality level ($CQoS_{w+1}$). Constraint (9) ensures that all required basic e-services to present Pg_k are supplied and finally, constraint (10) confirms that the selected suppliers have the capacity to supply this request.

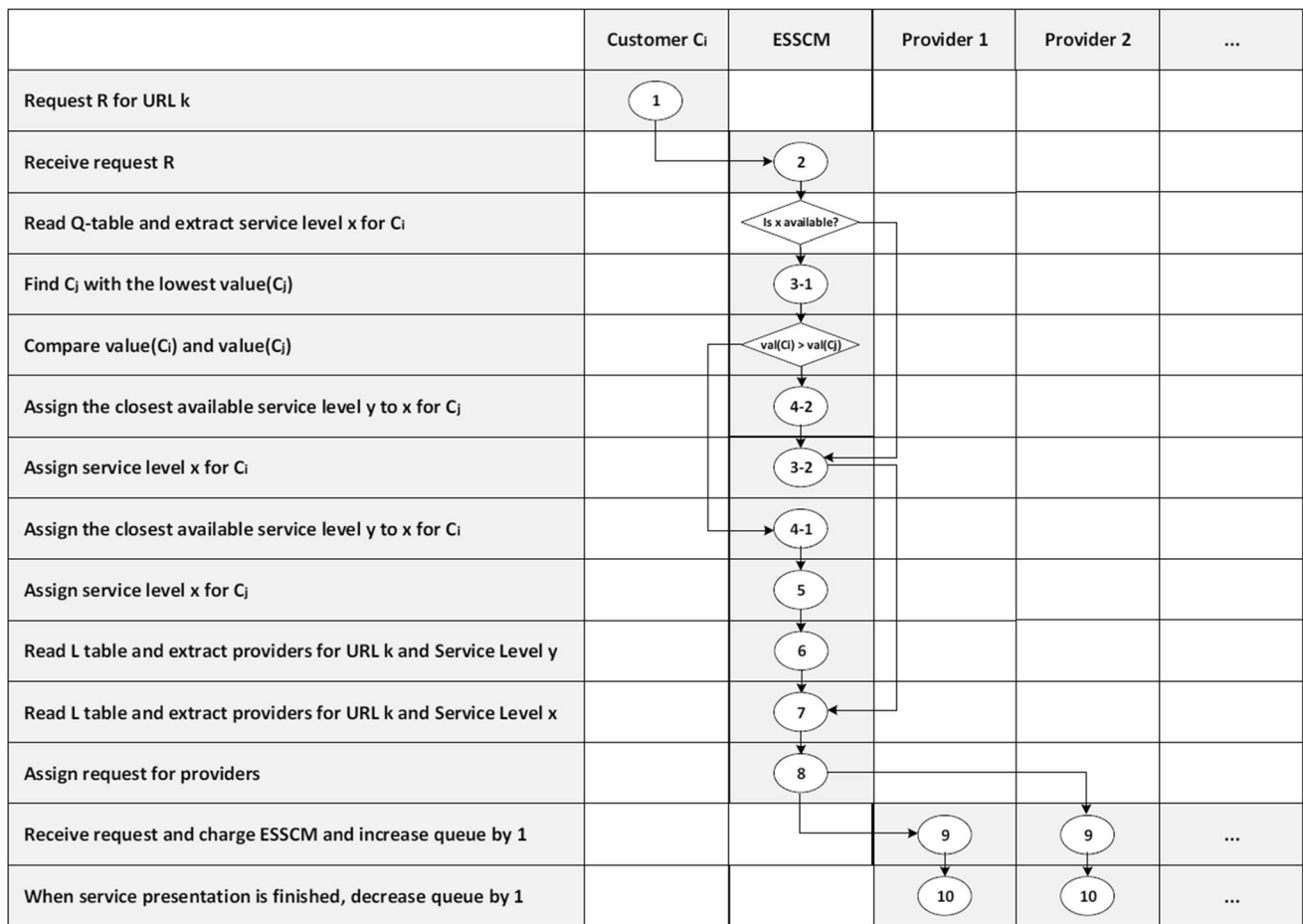|	Customer C_i	ESSCM	Provider 1	Provider 2	…
Request R for URL k	1				
Receive request R		2			
Read Q-table and extract service level x for C_i		Is x available?			
Find C_j with the lowest value(C_j)		3-1			
Compare value(C_i) and value(C_j)		val(Ci) > val(Cj)			
Assign the closest available service level y to x for C_j		4-2			
Assign service level x for C_i		3-2			
Assign the closest available service level y to x for C_i		4-1			
Assign service level x for C_j		5			
Read L table and extract providers for URL k and Service Level y		6			
Read L table and extract providers for URL k and Service Level x		7			
Assign request for providers		8			
Receive request and charge ESSCM and increase queue by 1			9	9	…
When service presentation is finished, decrease queue by 1			10	10	…

Fig. 5 The process of adaptive management of the e-commerce supply chain

Note that the supplier selection decision is not made in each request. Because the optimized combination of e-service providers is computed and is initially inserted into the L-table by the ECSCM agent and the L-table's content remains invariant until the situation of at least one e-service provider changes, for example one of the down e-service providers becomes available or vice versa. Therefore, the computational time for solving the MIP problem is not a serious concern in this work.

4.2.7 Adaptive Management of the E-Commerce Supply Chain

In this step, the ECSCM agent using the Q-Table for determining the appropriate QoS level and the L-table for choosing the basic e-service providers is able to manage the supply chain adaptively. The process of adaptive responding of the ECSCM agent to the customers' requests is illustrated below (Fig. 5).

In this process, upon receipt of the customer (c_i) request, the ECSCM agent reads the Q-table and according to the current state of the customer in his/her navigation extracts the appropriate QoS level (x). If x is available, i.e., if at least one combination of basic e-services providers to supply x exists, then the ECSCM agent by reading the L-table finds the optimized combination of basic e-services providers and allocates the request to them. If x is not available, then the ECSCM agent looks for the customer with the lowest value (c_j) who is receiving x right now. The value of each customer $c_k(Val(c_k))$ in his/her navigation state is defined as $Val_{n_{lc}}$ where $c = c_k$. If $Val(c_i) > Val(c_j)$, then the ECSCM agent replaces (c_j) with (c_i) and assigns the closest available QoS level (y) to (c_j). If $Val(c_i) \leq Val(c_j)$, then the ECSCM agent assigns y to (c_i). After determining the QoS level, the ECSCM agent reads the L-table and finds the optimized combination of basic e-services providers and allocates the request to them. The basic e-services providers after receipt of the request present the basic e-services and charge the ECSCM agent.
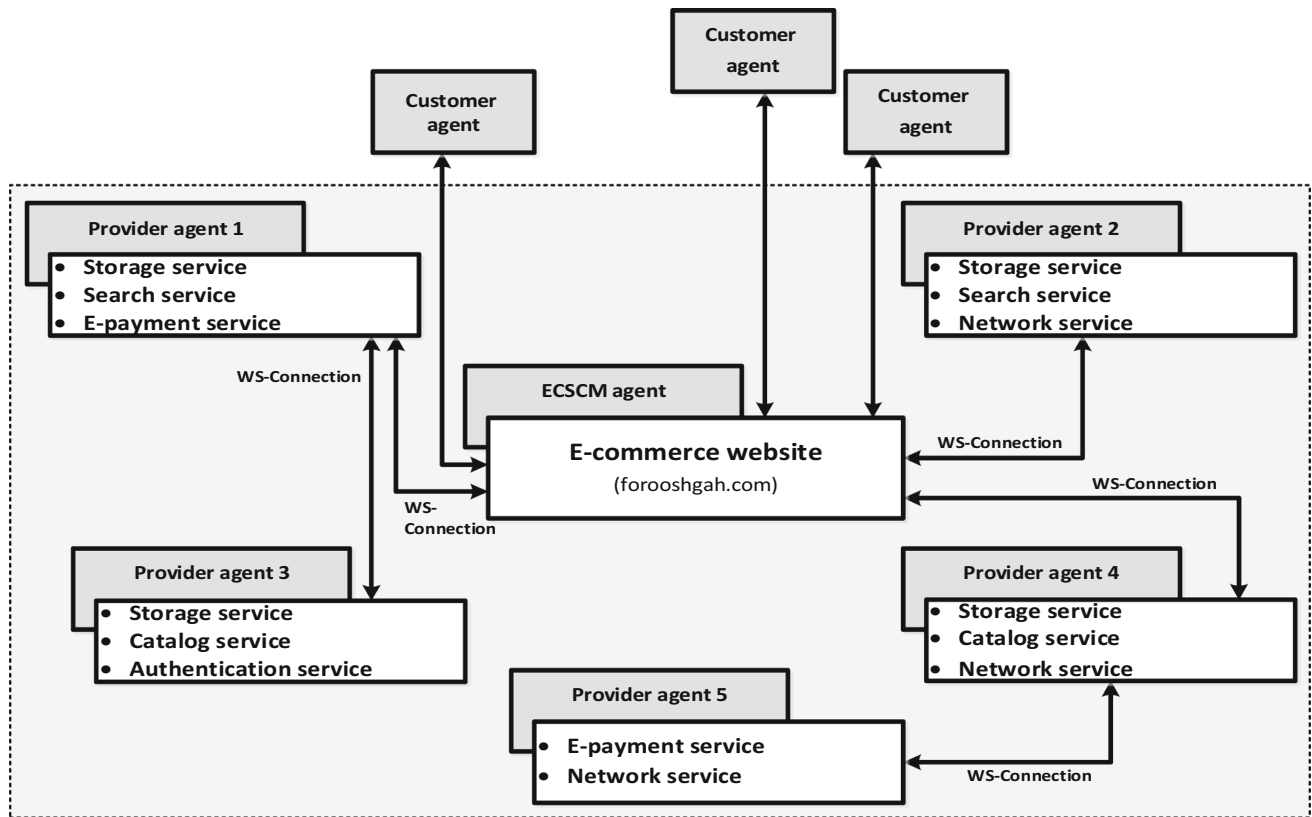
Fig. 6 The platform of the experiment system

5 Evaluations

5.1 Simulation Setup

This study designs an experimental environment based on a multi-agent system that is similar to the real world e-commerce supply chain environment. Due to the capacity of agent technology to develop complex and dynamic distributed systems, this is an appropriate paradigm to simulate the interactive relations between customers and the components of a e-commerce supply chain system.

There are four different types of agents in the considered multi-agent system. There is (1) the buyer agent that indicates a customer who navigates the e-commerce Website according to QoS-CBMG for buyers. Also there is (2) the Browser agent that indicates a visitor customer who navigates the e-commerce Website according to QoS-CBMG for browsers. The QoS-CBMGs for buyer and browser agents are constructed using the server log of real customers of the e-commerce Website. A detailed methodology of extracting QoS-CBMGs from real server logs is proposed in Al-Masri and Mahmoud (2007). Furthermore there is (3) the Provider agent that indicates a basic e-service provider who supplies different kinds of basic e-services with determined QoS levels and supply

capacity. In this paper, it is assumed that for the basic e-services supplied by each provider agent, the prices and capacities are invariant. Finally, there is (4) the ECSCM agent that selects the QoS level and basic e-services providers to respond to each customer's request according to the method as mentioned in Sect. 4.

The simulations were executed in the java agent development framework (JADE). JADE is a Java-based FIPA-compliant (Foundation of Intelligent Physical Agents) middleware that supports the development of distributed multi-agent applications based on the peer-to-peer communication architecture (Bellifemine et al. 2001).

In the experiment system, as shown in Fig. 6, there is an e-commerce Website that is managed by an ECSCM agent, and there are five basic e-service provider agents that provide different Web services, such as storage, media, e-payment, authentication, search and network services with different QoS levels. Additionally, there are several buyer and browser customer agents that navigate the Website and send requests to the ECSCM agent. The agents in this simulation environment interact with each other according to the flow diagram presented in Fig. 7.

A screen shot of the proposed adaptive supply chain system under execution in JADE is illustrated in Fig. 7. It shows various messages relating to interaction protocols
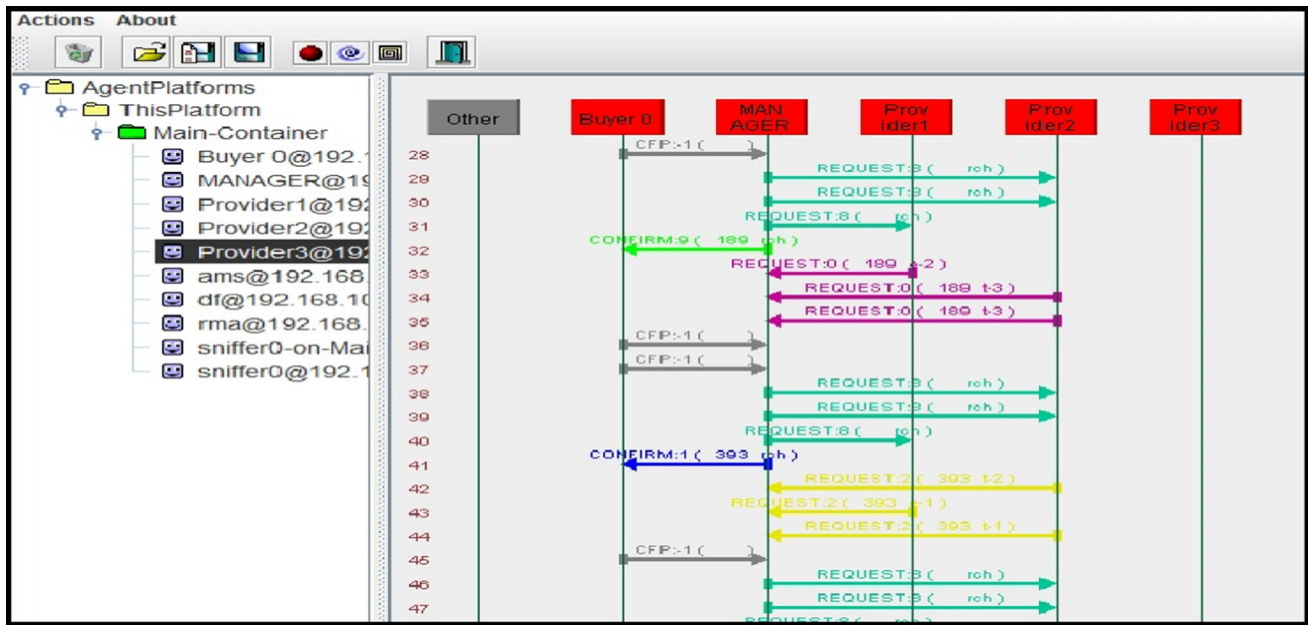
Fig. 7 A screen shot of the model in JADE

being exchanged between the agents comprised by the system.

The navigation of a customer agent on the e-commerce Website sends requests to the server. A sample system log of the simulated system is illustrated in Table 4. In the system log, the customers' requests and the allocated QoS level and providers are recorded. For example, in the table below, the first request is received from Browser 0 (a browser customer agent) in the Browse state of QoS-CBMG. ECSCM according to its Q-table allocates CQoS level 2 to this request and, based on its L-table, selects

providers 3, 4 and 1 to provide catalog, network and storage services with QoS levels 2, 2 and 2, respectively.

To evaluate the effect of the proposed adaptive model on the profitability and utility issues of the e-commerce system, we construct two versions of the above simulation platform. In the first version, it is assumed that only one composed QoS level is available and the requests are responded to in the order of arrival (First-In-First-Out strategy), and the second version utilizes the proposed model to assign different QoS levels and providers for each customer agent's request. The results obtained for designed

Table 4 A sample log of the simulated system

Customer ID	Requested URL	Date	Time	Adapted CQOS	Provider	Service	QoS	Provider	Service	QoS	Provider	Service	QoS
Browser 0	Browse	2015/02/17	03:55:33	2	Provider3	Catalog	2	Provider4	Network	2	Provider1	Storage	2
Buyer 1	Browse	2015/02/17	03:55:33	1	Provider4	Catalog	2	Provider2	Network	1	Provider1	Storage	1
Buyer 0	Search	2015/02/17	03:55:33	3	Provider1	Search	3	Provider2	Network	3	Provider1	Storage	2
Browser 0	Exit	2015/02/17	03:55:35	1									
Browser 0	Browse	2015/02/17	03:55:35	1	Provider1	Catalog	2	Provider2	Network	1	Provider1	Storage	1
Buyer 1	Exit	2015/02/17	03:55:37	1									
Buyer 1	Browse	2015/02/17	03:55:37	1	Provider3	Catalog	2	Provider2	Network	1	Provider1	Storage	1

Table 5 The parameters of the experiment scenarios

Parameter		The variation range	Number of experiments
Number of provider agents		2-6	5
Service criteria for each provider	Number of provided services	6	
	QoS levels	1-3	
	Prices	Low-Medium-High	
	Capacities	Low-Medium-High	
Average arrival rate of customer agents		100-600	6
Percentage of buyer agents		5-20	5

experiment scenarios from the various assessment indicators are presented in the next subsection.

5.2 Experiments

There are different parameters related to the agents and the Q-learning algorithm that are configured in experimental scenarios as explained in the following. Customer agent arrivals follow the Poisson distribution with average arrival rates ranging from 100 to 600 users per hour. Some of the customer agents are buyers and the others are browsers that navigate the Website according to QoS-CBMGs for buyers and browsers, respectively. The percentage of buyers has been changed from 5 to 20% of the total customer agents. For convenience, it is assumed that the e-commerce Website sells products/services with identical price (e.g., everything is $5) and customer agents can buy only one product/service at a time. The basic e-services provided by each provider agent, all basic e-service available QoS levels, prices and capacity constraints for each provider agent have been changed as shown in Table 5.

When starting the Q-learning algorithm, the values of the state-action pairs, $Q(s, a)$ can be initialized arbitrarily or assigned specific relative values to represent the preference in favoring each possible alternative. In this study, all the values of the state-action pairs are initially set to zero since all the actions for each state are assumed to be an equally valid choice. This approach starts the system from a neutral state assuming no a priori knowledge of which QoS allocation strategy is best to use in any situation. Therefore, the system is required to learn from scratch. The value of the discount-rate parameter γ can be set between zero and one. As γ approaches zero, the agent is more myopic because it takes immediate reward into account more strongly. On the other hand, as γ approaches

1, the agent will more farsightedly reduce the impact that recent results have on the learned policy (Ghavamipoor and Golpayegani 2017). Several example systems, such as those illustrated in Ghavamipoor and Golpayegani (2017) apply the Q-learning algorithm with a setting of $\gamma = 0.9$. This study uses the same common parameter settings for all the experimental runs. The simulation was allowed to run until 50,000 episodes were executed in each experimental scenario.

5.2.1 Assessment Indicators

To evaluate the proposed model, the above experiments are executed on both the proposed and the baseline model. The baseline model has the same structure of e-commerce supply chain as the proposed model, but the assignment of QoS levels and basic service providers to the customer requests are done randomly. The results of the evaluation are presented from two viewpoints: profitability and efficiency. To analyze the effect of the model on profitability, three assessment indicators are measured: (1) total profit, (2) total costs, and (3) total revenue. To measure efficiency, three assessment indicators are considered: (1) utilization of providers, (2) the frequency of lack of service occurrence, and (3) the average QoS level which is offered to the customers.

5.2.2 The Obtained Results

In the following, the evaluation results for the designed experiments are presented. As shown, the total profit of the e-commerce Website, which is our main indicator, improves in the experiment scenarios. The average of profit enhancement in the scenarios related to the arrival rate of customer agents, percentages of buyer agents, and the

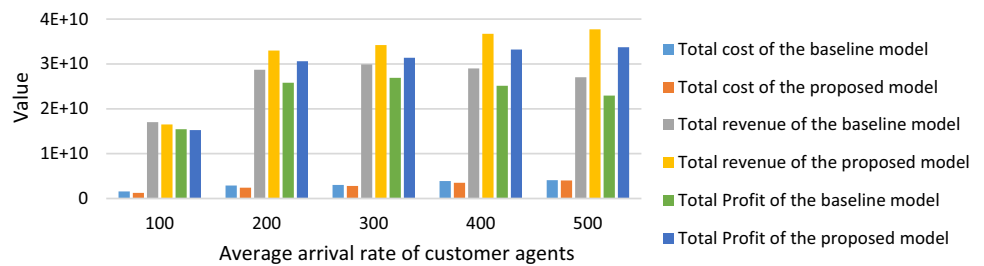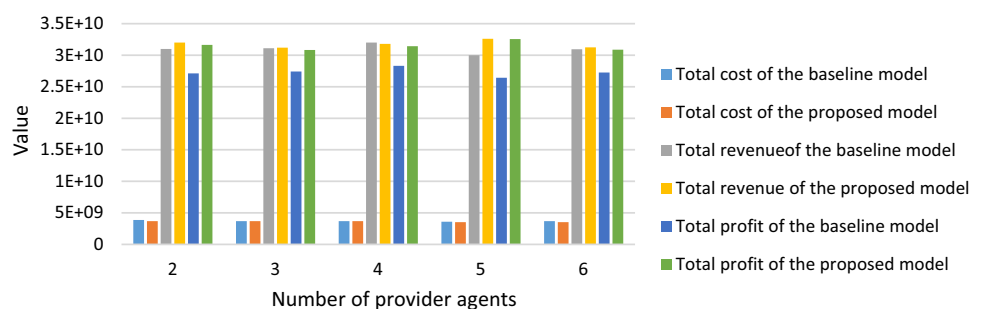**Fig. 8** Profitability analysis results of the scenarios related to the arrival rate of customer agents

Fig. 9 Profitability analysis results of the scenarios related to the percentages of buyer agents

Fig. 10 Profitability analysis results of the scenarios related to the number of provider agents

number of provider agents are 22.64%, 18.42% and 15.33%, respectively. The results presented in Figs. 8, 9 and 10 show that the total profit improvement is based on both the cost reduction and the revenue increase which occurs in the proposed adaptive model. As indicated in the following figures, on average a 4.4% cost reduction and 11.2% revenue increase are obtained.

In Fig. 8, the total profit of the base line model decreased when the arrival rate of customers became greater than 300 per hour. Because at this threshold (300 visitors per hour), the customers in the base line model experienced a low QoS level which led to dropout. But in the proposed model, the QoS levels which the customers received in the experiments with 400 and 500 arrivals per hour were tolerable for them and did not lead to a decline in the total profit, though in these experiments the rate of profit growth was reduced.

As shown in Fig. 9, the total profit of the proposed model is higher than that of the baseline model for a different percentage of buyer agents. In this figure, when the

percentage of buyer agents increased from 15 to 20%, the total profit in the proposed model remained almost stable, even though the total revenue grew due to more purchases. The reason is the increase of supplying costs, because the proposed model provided better service quality for buyers which contributed to higher supplying costs.

Furthermore, as illustrated in Fig. 10, when the number of provider agents grows the total profit of the proposed model becomes higher than that of the base line model, because with increasing the number of providers, the diversity of available QoS levels increases and the proposed model is able to apply its service differentiation mechanism.

As shown in Figs. 11, 12 and 13, the utilization of the providers is improved in 64% of the scenarios. Especially in cases where there are sufficient suppliers for a variety of QoS levels, we obtained up to approximately 80% enhancement of provider utilization in some scenarios. In some cases, the imbalanced distribution of requests among providers leads to a decrease of this indicator. The results
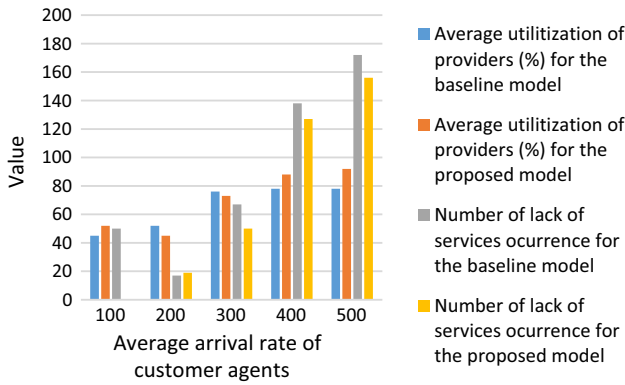
Fig. 11 Efficiency analysis results of the scenarios related to the arrival rate of customer agents
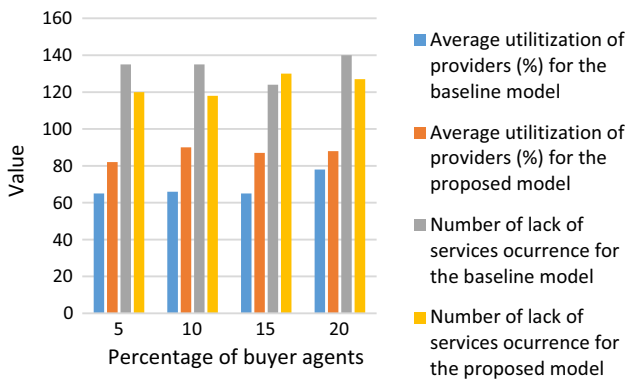

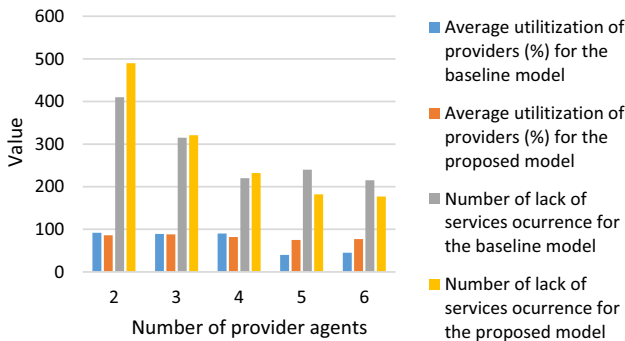**Fig. 12** Efficiency analysis results of the scenarios related to the percentages of buyer agents

Fig. 13 Efficiency analysis results of the scenarios related to the number of provider agents

Fig. 14 Average of the offered QoS level considering different arrival rates of customer agents

Fig. 15 Average of the offered QoS level considering different percentages of buyer agents

Fig. 16 Average of the offered QoS level considering different numbers of providers

for the second efficiency indicator (the occurrence frequency of the lack of services) are similar to the providers' utilization factor. The lack of QoS level diversity in provided basic e-services disturbs the balance of the requests distribution in the proposed model.

As these figures illustrate, in almost all scenarios the lack of services for the proposed model occurs less often than the base line model. But in Fig. 13, where the number
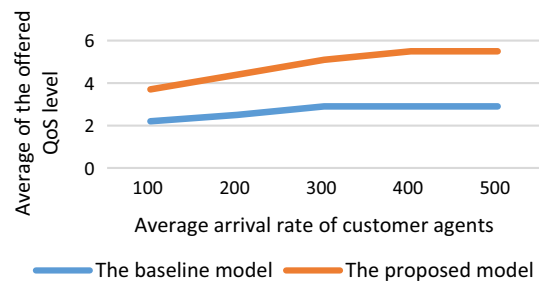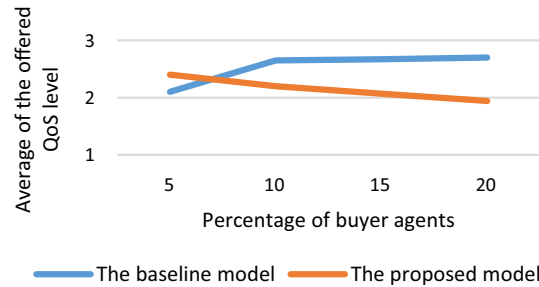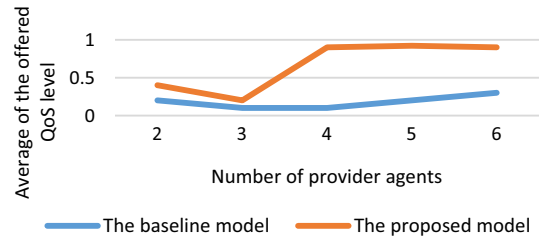
of the provider agents was low and consequently the diversity of QoS levels was low, the lack of services increased in the proposed model, because the distribution of requests between providers became imbalanced.

As the last efficiency indicators in Figs. 14, 15 and 16 show, the average QoS level which is offered to customers improved in almost all scenarios. This improvement is very important in e-commerce systems because response time, as a QoS measure in this study, plays an important role in customer satisfaction. According to Ghavamipoor and Golpayegani (2016) the perceived response time to a customer is a decisive factor in whether he/she will purchase on a Website or even return to it.

Table 6 The experimental setup

Parameter		The variation range	Number of experiments
Number of provider agents/cloud resource agents[*]		2-6	5
Service criteria for each provider /cloud resource	Number of provided services[*]	6	
	QoS levels	1-3	
	Prices	Low-Medium-High	
	Capacities*	Low-Medium-High	
Average arrival rate of customer agents		100-600	6
Percentage of buyer agents		5-20	5

Table 7 Comparison of the proposed model with the related works

References	The percentage of profit improvement (%)
Brusilovsky et al. (2007)	7
Chen et al. (2015)	11
Chen and Ryu (2013)	12
The proposed model	18.6

Table 8 Summary of the obtained results

Indicator	Average of improvement percentage in comparison with the baseline model (%)
cTotal cost	4.4
Total revenue	11.2
Total profit	18.6
Utilization of providers	20.5
Number of lack of services occurrence	10.18
The average of offered QoS level	115.15

5.3 Comparison

In addition to the baseline model, the proposed model was compared with some related admission control models (Totok and Karamcheti 2010; Lakshmi et al. 2017; Poggi 2014) regarding the profit aspect, as this is the main objective of the proposed model. For this purpose, the following experiment setup was considered (Table 6). Table 6 provides the same setup of the parameters as Table 5, but given the fact that in Lakshmi et al. (2017) cloud resources take the place of basic e-service providers, the provider-related parameters including "number of provider agents" and "service criteria for each provider" are allocated to cloud resources. In addition, considering that in Totok and Karamcheti (2010) and Poggi (2014) the models were proposed for an e-commerce Website, the parameters related to the basic providers (marked with asterik) are eliminated from the experiment scenarios of these models. Also the QoS levels and prices are assigned to the e-commerce Website provider. As illustrated in Table 6, these models are compared regarding different ranges of QoS levels, prices, arrival rates of customers and the percentage of buyers.

The average profitability improvement percentages of the entire experiments are provided in Table 7. Based on

the obtained results, the proposed adaptive model had a more positive effect on the profitability of the e-commerce system than the other models.

6 Conclusions and Future Work

This paper introduced an RL-based model for adaptive management of an e-commerce supply chain. At the core of the proposed model is an ECSCM agent that attempts to maximize the total profit of the e-commerce system through its two main functions: QoS-adaptation and service provider selection. The distributed architecture of the proposed framework facilitates its application in cloud-based service-supplying environments, which are increasing. As the experimental results show, some improvements in both profitability and efficiency were obtained particularly in situations where a sufficient variety of service providers and presentable QoS-levels were available. The following table (Table 8) summarizes the results obtained from the experiments.

As shown in Table 7, the profit enhancement of the proposed model is rather due to revenue increase than to cost reduction. Therefore, the proposed model is successful in encouraging the customers to navigate the purchase pattern. The results indicate that the average QoS level offered to the customers increases significantly in our model. The service quality level presented to the customers is directly related to their satisfaction (Poggi et al. 2011). Therefore, one of the important advantage of the model is improving customer satisfaction.

Nowadays, there is a wide range of external e-service providers (such as cloud providers) that present different basic services with a variety of QoS levels and prices. In this situation, the proposed model enables e-commerce providers to adapt QoS levels for different classes of customers, utilizing the advantage of this diversity of the basic e-service providers and leading to an improvement of the Website's profitability. At the same time, the proposed model provides better service quality for customers in average, so it can boost their overall satisfaction and return rate (Poggi et al. 2011).

Since the proposed model presents a general framework for adaptive e-service supply chain management, in future studies this framework can be utilized for different types of e-services, such as e-learning and e-banking, with different optimization goals, such as maximizing system performance, maximizing user satisfaction, and minimizing resource usages. Additionally, the proposed model can be extended with regard to additional adaptive features related to the Website's structure and the Website's presentation.

References

Aciar S, Zhang D, Simoff S, Debenham J (2007) Informed recommender: basing recommendations on consumer product reviews. IEEE Intell Syst 22(3):39–47

Al-Masri E, Mahmoud QH (2007) QoS-based discovery and ranking of web services. In: 16th International conference on computer communications and networks. IEEE, pp 529–534

Bathumalai G (2008) Self-adapting websites: mining user access logs. Dissertation, Robert Gordon University, Aberdeen

Belk M, Fidas C, Germanakos P, Samaras G (2015) Do human cognitive differences in information processing affect preference and performance of CAPTCHA? Int J Hum Comput Stud 84:1–18

Bellifemine F, Poggi A, Rimassa G (2001) JADE: a FIPA2000 compliant agent development environment. In: 5th International conference on autonomous agents. ACM, pp 216–217

Bhatti N, Friedrich R (1999) Web server support for tiered services. IEEE Netw 13(5):64–71

Brusilovsky P, Kobsa A, Nejdl W (2007) The adaptive web. Lecture notes in computer science, vol 4321. Springer, Cham, pp 325–341

Chan NN, Gaaloul W, Tata S (2012) A recommender system based on historical usage data for web service discovery. Serv Oriented Comput Appl 6(1):51–63

Chen M, Ryu YU (2013) Facilitating effective user navigation through website structure improvement. IEEE Trans Knowl Data Eng 25(3):571–588

Chen CC, Huang TC, Park JJ, Yen NY (2015) Real-time smartphone sensing and recommendations towards context-awareness shopping. Multimed Syst 21(1):61–72

Ewing JM, Menascé DA (2009) Business-oriented autonomic load balancing for multi-tiered Web sites. In: IEEE international symposium on modeling, analysis and simulation of computer and telecommunication systems. IEEE, pp 1–10

Ghavamipoor H, Golpayegani SAH (2013) Comparing and applying the approach of supply chain in electronic services management. Int J Comput Inf Technol 1(2):118–136

Ghavamipoor H, Golpayegani SAH (2016) QoS-aware provider selection in e-services supply chain. In: 8th International conference on information and knowledge technology. IEEE, pp 258–262

Ghavamipoor H, Golpayegani SAH (2017) A QoS sensitive model for e-commerce customer behavior. J Res Interact Mark 11(4):380–397

Harini N, Padmanabhan TR (2013) Admission control and request scheduling for secured-concurrent-available architecture. Int J Comput Appl 63(6):24–30

Hong J, Suh EH, Kim J, Kim S (2009) Context-aware system for proactive personalized service based on context history. Expert Syst Appl 36(4):7448–7457

Lakshmi MS, Kumar SP, Janardhan M, Gayathri K (2017) Machine learning methods for refining SLA based admission control and resource allocation in cloud computing. Int J Adv Res Comput Sci 8(9):834–840

Larisa G, Mariia S, Andriy R (2014) Control strategy of the input stream on the online charging system in peak load moments. In: 24th International crimean conference microwave and telecommunication technology. IEEE, pp 312–313

Lee Y, Kozar KA (2006) Investigating the effect of website quality on e-business success: an analytic hierarchy process (AHP) approach. Decis Support Syst 42(3):1383–1401

Li K, Jamin S (2000) A measurement-based admission-controlled web server. In: Proceedings of the 19th annual joint conference of the IEEE computer and communications societies, vol 2, pp 651–659

Li YM, Wu CT, Lai CY (2013) A social recommender mechanism for e-commerce: combining similarity, trust, and relationship. Decis Support Syst 55(3):740–752

Lin HF (2007) The impact of Website quality dimensions on customer satisfaction in the B2C e-commerce context. Total Qual Manag Bus Excell 18(4):363–378

Mark K, Csaba L (2007) Analyzing customer behavior model graph (CBMG) using Markov chains. In: 11th International conference on intelligent engineering systems. IEEE, pp 71–76

Menascé DA (2002) QoS issues in web services. IEEE Internet Comput 6(6):72–75

Poggi N (2014) AUGURES: profit-aware web infrastructure management. Doctoral thesis, Polytechnic University of Catalonia

Poggi N, Carrera D, Gavalda R, Ayguadé E (2011) Non-intrusive estimation of QoS degradation impact on e-commerce user satisfaction. In: 10th IEEE international symposium on network computing and applications. IEEE, pp 179–186

Raufi B, Georgieva J, Luma A, Ismaili F, Zenuni X (2010) An adaptation algorithm for adaptive Web based systems based on link structure and document similarity. In: 9th WSEAS international conference on telecommunications and informatics. World Scientific and Engineering Academy and Society, pp 29–34

Rosaci D, Sarné GM (2012) A multi-agent recommender system for supporting device adaptivity in e-commerce. J Intell Inf Syst 38(2):393–418

Sarwar B et al (2001) Item-based collaborative filtering recommendation algorithms. In: 10th international conference on world wide web. ACM, pp 285–295

Schafer JB et al (2007) Collaborative filtering recommender systems. The adaptive web. Springer, Heidelberg, pp 291–324

Suchacka G, Borzemski L (2013) Web server support for e-customer loyalty through QoS differentiation. In: Nguyen NT (eds) Transactions on computational collective intelligence XII. Lecture Notes in Computer Science. vol. 8240. Springer, Heidelberg

Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge

Totok A, Karamcheti V (2010) RDRP: reward-driven request prioritization for e-commerce web sites. Electron Commer Res Appl 9(6):549–561

Urgaonkar B (2005) Dynamic resource management in internet hosting platforms. Doctoral dissertation, University of Massachusetts Amherst

Urgaonkar B, Shenoy P (2005) Cataclysm: policing extreme overloads in internet applications. In: 14th International conference on world wide web. ACM, pp 740–749

Weiss G (1999) Multiagent systems: a modern approach to distributed artificial intelligence. MIT Press, Cambridge

Yin PY, Guo YM (2013) Optimization of multi-criteria website structure based on enhanced tabu search and web usage mining. Appl Math Comput 219(24):11082–11095

Yue C, Wang H (2007) Profit-aware admission control for overload protection in e-commerce web sites. In: 15th IEEE international workshop on quality of service. IEEE, pp 188–193

Zatwarnicki K, Zatwarnicka A (2014) The cluster-based time-aware web system. In: International conference on computer networks. Springer, pp 37–46

Zheng Z, Zhang Y, Lyu MR (2014) Investigating QoS of real-world web services. IEEE Trans Serv Comput 7(1):32–39

Springer