Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A strategy-proof auction mechanism for service composition based on user preferences*

Yao XIA, Zhiqiu HUANG‡

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

E-mail: xiayao@nuaa.edu.cn; zqhuang@nuaa.edu.cn

Received Dec. 23, 2019; Revision accepted June 30, 2020; Crosschecked Dec. 11, 2020

Abstract: Service composition is an effective method of combining existing atomic services into a value-added service based on cost and quality of service (QoS). To meet the diverse needs of users and to offer pricing services based on QoS, we propose a service composition auction mechanism based on user preferences, which is strategy-proof and can be beneficial in selecting services based on user preferences and dynamically determining the price of services. We have proven that the proposed auction mechanism achieves desirable properties including truthfulness and individual rationality. Furthermore, we propose an auction algorithm to implement the auction mechanism, and carry out extensive experiments based on real data. The results verify that the proposed auction mechanism not only achieves desirable properties, but also helps users find a satisfactory service composition scheme.

Key words: Combinatorial reverse auction; Service composition; User preference; Strategy-proof; Dynamic pricing
https://doi.org/10.1631/FITEE.1900726 **CLC number:** TP311.5

1 Introduction

The prominent features of distributed computing have rapidly driven the development of cloud computing. Service providers have provided many functionality-equivalent services with different prices and quality-of-service (QoS) levels. To fulfill the user's complex business requirements, they need to compose the atomic services into a powerful new service. Because every service has multiple attributes, such as response time, availability, and reliability, the service composition problem is a multi-attribute decision-making problem, which leads to great challenges in selecting preferred services from a wide range of services.

Many existing studies on QoS-aware service composition are based on the fixed pricing model (i.e., pay as you go). In this charging model, the user cannot evaluate the true cost of a service, because the service price is determined by the service providers. However, there is a deficiency in this charging model. On one hand, service providers may lie about the true cost (i.e., minimum cost) of a service, because they are self-interested and wish to maximize their profits. On the other hand, price cannot dynamically reflect the supply and demand in the cloud market. It is difficult for users to select satisfactory services. There is a need to design an incentive mechanism to motivate service providers to report the true cost of services.

Auction is a popular dynamic pricing mechanism that is widely used in cloud computing, such as Amazon's cloud computing resource allocation mechanism, due to its high efficiency and fairness (Wu QW et al., 2018). Recent studies (Tanaka and Murakami, 2016; Wu QW et al., 2018) applied the Vickrey-

‡ Corresponding author

* Project supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization, the National Key Research and Development Program of China (Nos. 2016YFB1000802 and 2018YFB1003900), and the National Natural Science Foundation of China (No. 61772270)

ORCID: Yao XIA, https://orcid.org/0000-0001-5551-2570; Zhiqiu HUANG, https://orcid.org/0000-0001-6843-1892

Clark-Groves (VCG) approach to the service composition problem. Although it can solve the issue of untruthful service cost, the services' QoS at the winner determination and payment calculation stage is not considered. In their studies, they first selected services that met the user's QoS constraints, and then determined the winning bids and calculated their prices based on the reported costs of services in the rest of the service set. However, because users have diverse QoS preferences, it is difficult to design winner determination and pricing by considering user preferences in the auction mechanism, and to select services that the user prefers. Therefore, it is essential to design a new auction mechanism to compose services based on user preferences and satisfy the user with diversified choices. The proposed mechanism can motivate service providers to offer services at lower prices with high QoS. However, there are no studies in which this mechanism is implemented.

In this study, we target at designing an auction mechanism for finding the global service composition scheme that users prefer and dynamically pricing the service. However, the following challenges exist when designing a practical auction mechanism for the service composition problem: (1) How can we design a winner determination mechanism to select winning bids that users prefer? (2) How can we design the payment calculation mechanism to guarantee that service providers report the true cost of services while considering user preference? (3) How can we ensure that the auction mechanism achieves the desirable properties of truthfulness and individual rationality?

To deal with these challenging issues, we propose a service composition auction mechanism named SCAUP, which is based on user preferences and is strategy-proof. SCAUP provides a platform such that the user procures services from service providers and service providers compete to sell services to the user. In this mechanism, user preferences are considered in both winner determination and payment calculation. Moreover, it can motivate service providers to report the true cost of services and dynamically determine the price of services according to the QoS and the reported cost of service providers. Finally, we validate that SCAUP achieves truthfulness and individual rationality through theoretical proof and extensive experiments.

2 System model

In this section, we introduce the auction-based service composition problem and the auction model. Important notations used in this paper are shown in Table 1.

Table 1 Important symbols

Symbol	Description
as_i	The i^{th} atomic service in an abstract service composition
B_i	Bid set of abstract service as_i
B	Set of all bids
u	Utility of the user
$b_{i,j,k}$	The j^{th} bid of as_i provided by service provider k
WS	Set of winning bids
$\mathrm{vc}_{i,j,k}$	Virtual cost of $b_{i,j,k}$
$c_{i,j,k}$	Reported cost of $b_{i,j,k}$
$p_{i,j,k}$	Price of $b_{i,j,k}$
usp_k	Utility of service provider k
ω_d	Weight value of the d^{th} attribute
$q_{i,j,k,1}$	Reported response time of $b_{i,j,k}$
$q_{i,j,k,2}$	Reported availability of $b_{i,j,k}$
$q_{i,j,k,3}$	Reported reliability of $b_{i,j,k}$
RT_i	Minimum response time of service as_i
A_i	Maximum availability of service as_i
R_i	Maximum reliability of service as_i
$\mathrm{WS}^{i,j,-k}$	Set of winning bids except $b_{i,j,k}$ from service provider k
WS^k	Set of winning bids of service provider k
W	Set of weight values of all QoSs

2.1 Problem statement

In cloud computing, there are many atomic services. To implement the complex workflow, the user needs to compose many atomic services. However, the user faces a decision problem when selecting the optimal service composition scheme (i.e., finding the service composition scheme with the minimum total cost and satisfactory QoS). Because a service has many attributes (e.g., cost, availability, response time, and reliability) and the user has preferences to service attributes, it is difficult for the user to select a satisfactory service.

We provide an auction platform for the service composition problem, where the user buys services from many service providers and composes these services based on the total cost and QoS. We consider the user's diverse needs for the service's attributes when he/she selects services. First, the user should design an abstract workflow and split it into many atomic services. The abstract service can be defined as WF={as_1, as_2, …, as_n}, where as_i (i=1, 2, …, n) is an atomic service. Then the user sends the service procurement specification (SPS) to the auctioneer. The service providers can submit bids to the auctioneer based on the SPS. Finally, the auctioneer determines the winning bid and calculates the price to be paid according to the rules of the auction mechanism.

2.2 Auction model

The auction model includes the user bidding model, service provider bidding model, and auctioneer model, involving a user, multiple service providers, and an auctioneer. We detail these three models in the following.

User bidding model: A user is a buyer who gets services from different service providers and designs the SPS for the auction. The formal definition is SPS=(WF, QoS, W, v, P), where the WF set is an abstract workflow and the QoS set defines the user's QoS constraints for each abstract service, that is, QoS={as_i, RT_i, A_i, R_i}, with RT_i, A_i, and R_i representing the minimum response time, maximum availability, and maximum reliability of specific service as_i, respectively, and P is the maximum total price for a service composition that the user can accept. Unlike other auction models, our model adds two parameters, W and v, to reflect the user preferences for QoS, W={ω_1, ω_2, …, ω_f} represents the user preferences for each QoS attribute. The user defines the preferred weight factors for each QoS such as response time, $\omega_d\in[0, 1]$, d=1, 2, …, f, with f referring to the total number of QoS attributes. In this study only three attributes are considered, so f=3. v is the adjustment factor of additional cost, affecting the proportion of the additional cost from QoS translation in the virtual cost; v expresses the user's QoS importance in payment calculation. The application of W and v in our auction mechanism will be detailed in Section 3.

Service provider bidding model: Service providers sell services to the user. Assume that r service providers participate in the auction, SP={1, 2, …, r}. Their bids are defined as $b_{i,j,k}$=(as_i, $q_{i,j,k,1}$, $q_{i,j,k,2}$, ..., $q_{i,j,k,d}$, $c_{i,j,k}$), where i=1, 2, …, n, k=1, 2, …, r, meaning that service provider k bids for abstract service as_i and promises to provide services with response time less than $q_{i,j,k,1}$, availability and reliability greater than $q_{i,j,k,2}$ and $q_{i,j,k,3}$, respectively, and the cost for the abstract service equal to $c_{i,j,k}$. B_i is the bid set of as_i; it can be represented as B_i={$b_{i,1,1}$, $b_{i,2,2}$, $b_{i,3,3}$, ..., $b_{i,j,k}$}, i=1, 2, …, n, j=1, 2, …, m, k=1, 2, …, r, where m is the number of bids for an abstract service. Bids for all abstract services can be expressed as B={B_1, B_2, …, B_n}.

Auctioneer model: An auctioneer is a truthful third party. It determines the set of winning bids, WS$\subseteq B$, and calculates the price of each bid, i.e., p={$p_{1,1,1}$, $p_{i,j,k}$, ..., $p_{n,m,r}$}, where $p_{i,j,k}$ is the payment to $b_{i,j,k}$, meaning that the j^{th} bid of as_i provided by service provider k wins the auction and obtains payment $p_{i,j,k}$.

We define the utility of service provider k as

$$\mathrm{usp}_k=\begin{cases}\sum\limits_{b_{i,j,k}\in \mathrm{WS}^k} p_{i,j,k}-\sum\limits_{b_{i,j,k}\in \mathrm{WS}^k} c_{i,j,k}, & \text{if } b_{i,j,k} \text{ wins,}\\ 0, & \text{otherwise,}\end{cases}\tag{1}$$

where WS^k denotes the winning bid set of service provider k.

Because the service composition problem is a combinatorial reverse auction, the user's utility is the difference between the maximum total price that the user is willing to pay and the sum price of all winning bids. The utility of the user can be defined as follows:

$$u=\begin{cases}P-\sum\limits_{b_{i,j,k}\in \mathrm{WS}} p_{i,j,k}, & \text{if } \sum\limits_{b_{i,j,k}\in \mathrm{WS}} p_{i,j,k}\le P,\\ 0, & \text{otherwise (lose auction).}\end{cases}\tag{2}$$

Definition 1 (Social welfare) The social welfare is the sum of the prices of the winning bids in the service composition scheme:

$$\mathrm{SW}=\sum_{b_{i,j,k}\in \mathrm{WS}} p_{i,j,k}.\tag{3}$$

2.3 Desirable properties

A practical auction mechanism should have the following properties:

Definition 2 (Truthfulness (Nisan et al., 2007; Zheng et al., 2015)) An auction mechanism is truthful, if and only if the dominant strategy for a service provider is to report services' truthful cost no matter what strategies other service providers choose. In other words, we assume that $s_{i,j,k}$ is the dominant strategy of service provider k. $\forall\ s'_{i,j,k} \neq s_{i,j,k}$ and the strategy profile of any other service providers $s_{i,-j,-k}$, we have $\mathrm{usp}_k\left(s_{i,j,k}, \overline{s_{i,-j,-k}}\right) \geq \mathrm{usp}_k\left(s'_{i,j,k}, \overline{s_{i,-j,-k}}\right)$, where $\overline{s_{i,-j,-k}}$ means the strategy profile set except the strategy of the j^{th} bid of service provider k for bidding service as_i.

Definition 3 (Individual rationality (Wen et al., 2015)) The user and service providers who participate in the auction obtain nonnegative utility, i.e., $\mathrm{usp}_k \geq 0$ and $u \geq 0$.

Definition 4 (Strategy-proof (Mu'Alem and Nisan, 2008; Singer, 2010; Zheng et al., 2017)) An auction mechanism is called strategy-proof if it has both truthfulness and individual rationality properties.

2.4 Auction process

The auction process includes the user, service providers, and the auctioneer (Fig. 1).

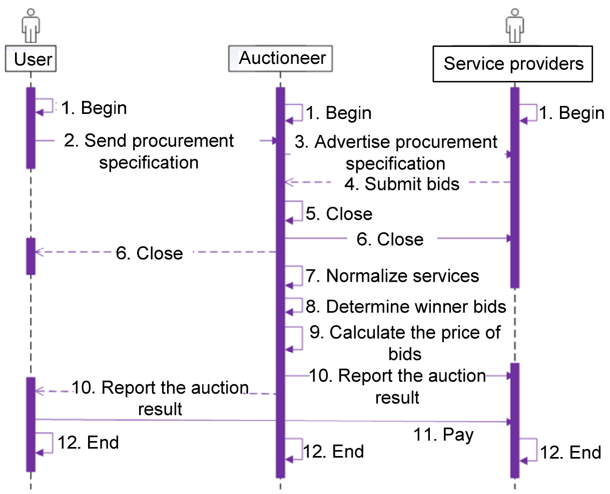

Fig. 1 Sequence of the proposed SCAUP mechanism

The auction sequence of SCAUP is as follows:

1. The service providers, user, and auctioneer start the auction.

2. Send the procurement specification: (1) The user needs to design the service composition model for the auction. The abstract workflow WF is first designed according to the user's business requirement and then decomposed into n abstract services, on which service providers can easily bid. (2) The user sends the procurement specification to the auctioneer.

3. The auctioneer advertises the user's procurement specification to the service providers.

4. Service providers prepare and submit their bids to the auctioneer.

5. The auctioneer closes the auction.

6. The service providers and the user close the auction.

7. The auctioneer normalizes the QoS attributes into [0, 1], and calculates the services' virtual cost for the winner determination and payment calculation phase. The attribute normalization and service virtual cost calculation methods are shown in Section 3.1.

8. The auctioneer determines the winner bids according to the winner determination mechanism.

9. The auctioneer calculates the price of each winning bid.

10. The auctioneer reports the auction result to the user and service providers.

11. The user pays the service providers for the selected services.

12. The entire auction process is completed.

3 Proposed SCAUP auction mechanism

In this section, we propose the SCAUP auction mechanism considering the user's service attribute preferences. In the VCG auction mechanism, winner determination and payment calculation are based only on the reported cost of services and the QoS is not considered, so service providers cannot be motivated to offer services with high-level QoS. Thus, it is essential to consider both the services' costs and QoS for the auction mechanism. Because services have many QoS attributes, such as response time, reliability, and availability, service selection is a multi-attribute decision-making problem. It is difficult for the user to choose the right services based on personal preferences. Therefore, it is necessary to design a virtual cost calculation method for a service to quantify and compare the prices and QoSs. We use the simple additive weighting method to obtain the total virtual cost of a service. The SCAUP auction mechanism has four parts: the QoS normalization and service virtual cost calculation methods, the winner determination mechanism, the price calculation mechanism, and the auction algorithm. First, the QoS normalization and service virtual cost calculation methods are detailed. Then, the design of winner determination and payment calculation is illustrated. Next, we prove that the SCAUP auction mechanism is strategy-proof. Finally, we present the auction algorithm to implement the SCAUP auction mechanism.

3.1 QoS normalization and service virtual cost calculation

Because the services provided by service providers have different QoS levels, and each QoS attribute has a different measurement unit, every QoS attribute value of a service must be normalized to a dimensionless quantity in interval [0, 1], which is an additional score. QoS includes positive (Q+) attributes and negative (Q^-) attributes. For a positive attribute, such as availability and reliability, the higher the value, the better the service performs. For a negative attribute, such as response time, the smaller the value, the better the service performs. The normalization methods for positive attributes and negative attributes are different.

3.1.1 QoS attribute normalization

For positive QoS attributes ($q_{i,j,k,d}\in Q^+$, i=1, 2, …, n, j=1, 2, …, m), Eq. (4) is used for normalization:

$$q'_{i,j,k,d}=\begin{cases}\dfrac{q_{i,j,k,d}-\min q_{i,d}}{\max q_{i,d}-\min q_{i,d}}, & \max q_{i,d}\neq \min q_{i,d},\\ 1, & \max q_{i,d}=\min q_{i,d}.\end{cases} \tag{4}$$

For negative QoS attributes ($q_{i,j,k,d}\in Q^-$, i=1, 2, …, n, j=1, 2, …, m), Eq. (5) is used for normalization:

$$q'_{i,j,k,d}=\begin{cases}\dfrac{\max q_{i,d}-q_{i,j,k,d}}{\max q_{i,d}-\min q_{i,d}}, & \max q_{i,d}\neq \min q_{i,d},\\ 1, & \max q_{i,d}=\min q_{i,d}.\end{cases} \tag{5}$$

Here n represents the total number of abstract services, m the total number of services corresponding to an abstract service, $q_{i,d}$ the set of $q_{i,j,k,d}$, and $\max q_{i,d}$ and $\min q_{i,d}$ the maximum and minimum values of the d^{th} attribute, respectively. After normalization, the value of each attribute $q'_{i,j,k,d}$ is within [0, 1], and a higher value indicates that the service performs better on the corresponding attributes.

In our auction mechanism $q'_{i,j,k,d}$ are considered as additional scores to evaluate the service's QoS attributes and user preferences. However, the evaluation criteria of a service's additional score must correspond to the evaluation criteria of subsequent winner determination and payment calculation. Since the reverse combination auction model established in this study determines the winner by the rule that the smaller the virtual cost, the better the result (see Section 3.2 for details), a transformation is carried out, $q^*_{i,j,k,d}=1-q'_{i,j,k,d}$ ($q_{i,j,k,d}\in$[0, 1]), to make the services with lower additional scores have better performance on the corresponding QoS attributes.

3.1.2 Service virtual cost calculation

The virtual cost of every service can be calculated using the simple additive weighting method. The weight value of each QoS attribute can be determined using the analytic hierarchy process (AHP) method and ultimately depends on user preferences. The virtual cost of $b_{i,j,k}$ can be calculated as

$$\mathrm{vc}_{i,j,k}=c_{i,j,k}+v\sum_{d=1}^{f}\omega_d q^*_{i,j,k,d}, \tag{6}$$

where $\sum_{d=1}^{f}\omega_d=1$, $\omega_d\in$[0, 1], d=1, 2, …, f, f refers to the total number of QoS attributes, $c_{i,j,k}$ is the reported cost of $b_{i,j,k}$, v is the adjustment factor of the additional cost set by the user, and $v\geq 0$. The second item on the right-hand side of Eq. (6) is the additional cost.

Based on the above algorithm, the smaller the virtual cost is, the better the overall performance of the service will be. When v=0, it means that the user no longer considers the additional QoS scores, and the virtual cost is the reported cost of the service.

3.2 Winner determination

The goal of QoS-aware service composition is to find the optimal service composition scheme based on user preferences. To solve this problem, we formulate the winner determination using 0–1 integer programming. Therefore, the objective function of winner determination is to minimize the total virtual cost of the service composition, which is equivalent to maximizing the overall performance of service composition. Different from other studies that consider only the costs of services, we consider both the costs of services and QoS in the winner determination mechanism. The proposed objective function is as follows:

$$\begin{aligned}&\min\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r}\mathrm{vc}_{i,j,k}x_{i,j,k}\\&=\min\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r}\left(c_{i,j,k}+v\sum_{d=1}^{f}\omega_d q^{*}_{i,j,k,d}\right)x_{i,j,k}\end{aligned}\tag{7}$$

$$\text{s.t.}\ q_{i,j,k,1}x_{i,j,k}\le \mathrm{RT}_i,\tag{7a}$$

$$q_{i,j,k,2}x_{i,j,k}\ge A_i,\tag{7b}$$

$$q_{i,j,k,3}x_{i,j,k}\ge R_i,\tag{7c}$$

$$\sum_{i=1}^{n}x_{i,j,k}\le n,\tag{7d}$$

$$x_{i,j,k}\in\{0,\ 1\},\tag{7e}$$

$$\omega_d\in[0,\ 1],\tag{7f}$$

where $x_{i,j,k}$ is the decision variable. When $x_{i,j,k}$=1, $b_{i,j,k}$ is selected; when $x_{i,j,k}$=0, $b_{i,j,k}$ is not selected.

3.3 Payment calculation

The proposed auction mechanism determines the winner according to the virtual cost of the service, which is the sum of the service cost and the additional cost of QoS conversion. There may be a situation where the service with a low additional cost and a high reported cost wins the bid, which means that the winner's reported cost is not the lowest. If payment is calculated based solely on the reported cost of the service, using the VCG payment calculation mechanism, it is possible to pay the winner less than its reported cost, which obviously does not guarantee individual rationality.

We propose a new payment calculation method based on the VCG mechanism. The difference is that we calculate the services' prices based on the virtual costs, considering the QoS and the reported cost, whereas VCG considers only the reported cost. We first calculate the total virtual cost of the service composition scheme (i.e., the WS set) when $b_{i,j,k}$ does not participate in the auction, and then calculate the total virtual cost of the service composition scheme when $b_{i,j,k}$ is not available. The price for each $b_{i,j,k}$ can be calculated as follows:

$$p_{i,j,k}=\sum_{b_{i',j',k'}\in\mathrm{WS}^{i,j,-k}}\mathrm{vc}_{i',j',k'}-\sum_{b_{i^*,j^*,k^*}\in\mathrm{WS},j^*\neq j}\mathrm{vc}_{i^*,j^*,k^*}.\tag{8}$$

3.4 Desirable property analysis

In this subsection we demonstrate that the SCAUP auction mechanism is strategy-proof.

Theorem 1 The SCAUP auction mechanism achieves truthfulness.

Proof The truthfulness of SCAUP is to be proved by distinguishing the following cases:

We assume $\mathrm{cost}_{i,j,k}$ is the true cost of $b_{i,j,k}$ when service provider k bids for service as_i. When $b_{i,j,k}$ honestly reports its true cost and wins the auction, it gains nonnegative utility.

There are two situations where $b_{i,j,k}$ dishonestly reports its cost:

1. The bid $b_{i,j,k}$ of service provider k reports a cost lower than $\mathrm{cost}_{i,j,k}$

If $b_{i,j,k}$ reports a cost lower than $\mathrm{cost}_{i,j,k}$ and the reported QoS is lower than the user's QoS constraint, $b_{i,j,k}$ loses the auction and the utility is zero, because all selected services must meet QoS constraints.

If $b_{i,j,k}$ reports a lower cost and a higher QoS than the user's QoS constraint, it may win the auction with the lowest virtual cost. However, the payment obtained by $b_{i,j,k}$ has not been increased, because its payment calculation is based on the virtual cost of other bids and is not affected by its own virtual cost, and its price does not change, so its utility does not change.

2. The bid $b_{i,j,k}$ reports a cost higher than $\mathrm{cost}_{i,j,k}$

If $b_{i,j,k}$ reports a cost higher than $\mathrm{cost}_{i,j,k}$ and the reported QoS is lower than the user's QoS constraint, it loses the auction and its utility is zero, because all selected services must meet the QoS constraint.

If $b_{i,j,k}$ reports a cost higher than $\mathrm{cost}_{i,j,k}$ and a QoS higher than the user's QoS constraint, it may still win the auction. However, because the payment of this bid is determined by the virtual cost of other bids and has nothing to do with its own virtual cost, the utility of $b_{i,j,k}$ is not to be improved.

As can be observed, $b_{i,j,k}$ cannot improve its utility by reporting higher or lower cost (i.e., untruthful cost). Therefore, the dominant strategy of the service providers is to report the true cost of the service, and the SCAUP auction mechanism achieves truthfulness.

Theorem 2 The SCAUP auction mechanism achieves individual rationality.

Proof If the service provider loses the auction, its utility is zero. We can then prove that the utility of the winning service provider is no less than zero:

$$\begin{aligned}\mathrm{usp}_k &= \sum_{b_{i,j,k}\in \mathrm{WS}^k} p_{i,j,k} - \sum_{b_{i,j,k}\in \mathrm{WS}^k} c_{i,j,k} \\ &= \sum_{b_{i,j,k}\in \mathrm{WS}^k}\left(\sum_{b_{i',j',k'}\in \mathrm{WS}^{i,j,-k}} \mathrm{vc}_{i',j',k'} - \sum_{b_{i^*,j^*,k^*}\in \mathrm{WS}, j^*\neq j} \mathrm{vc}_{i^*,j^*,k^*}\right) \\ &\quad - \sum_{b_{i,j,k}\in \mathrm{WS}^k} c_{i,j,k} \\ &\geq \sum_{b_{i,j,k}\in \mathrm{WS}^k}\left(\sum_{b_{i^*,j^*,k^*}\in \mathrm{WS}, j^*\neq j} \mathrm{vc}_{i^*,j^*,k^*} + \mathrm{vc}_{i,j,k}\right. \\ &\quad \left. - \sum_{b_{i^*,j^*,k^*}\in \mathrm{WS}, j^*\neq j} \mathrm{vc}_{i^*,j^*,k^*}\right) - \sum_{b_{i,j,k}\in \mathrm{WS}^k} c_{i,j,k} \\ &= \sum_{b_{i,j,k}\in \mathrm{WS}^k} \mathrm{vc}_{i,j,k} - \sum_{b_{i,j,k}\in \mathrm{WS}^k} c_{i,j,k} \geq 0.\end{aligned}$$

Note that because our goal in the above equations is to minimize the total virtual cost of the service composition, there is

$$\sum_{b_{i',j',k'}\in \mathrm{WS}^{i,j,-k}} \mathrm{vc}_{i',j',k'} \geq \sum_{b_{i^*,j^*,k^*}\in \mathrm{WS}} \mathrm{vc}_{i^*,j^*,k^*}.$$

Thus,

$$\sum_{b_{i',j',k'}\in \mathrm{WS}^{i,j,-k}} \mathrm{vc}_{i',j',k'} \geq \sum_{b_{i^*,j^*,k^*}\in \mathrm{WS}, j^*\neq j} \mathrm{vc}_{i^*,j^*,k^*} + \mathrm{vc}_{i,j,k}.$$

Finally, it can be obtained that usp_k is greater than or equal to zero.

Because SCAUP is a reverse auction, the user buys services from the service providers. If the best service composition scheme can be found, which makes the total price less than P, the user's utility $u>0$. If a scheme that meets the requirements cannot be found, the auction fails, and the user's utility $u=0$. So, the user's utility is nonnegative.

In conclusion, the SCAUP auction mechanism achieves individual rationality.

3.5 Auction algorithm design

We design an efficient auction algorithm to implement the SCAUP auction mechanism that can find the best service composition scheme and dynamically calculate the service's cost. The main components of the auction algorithm include service selection, winner determination, the particle swarm optimization (PSO) algorithm, and the payment calculation algorithm, which are described in the following.

3.5.1 Service selection

Because the user declares the constraint conditions of QoS attributes in the procurement specification, the service selection algorithm is designed to select those services that conform to the user's QoS attribute constraint. The pseudocode is given in Algorithm 1.

Algorithm 1 ServiceSelection(B, r, n)

```
Input: B
N: number of B's
r: number of service providers
n: number of abstract services
AS1=∅
AS=∅
Output: AS set
for (k=1, 2, …, r) do
   for (j=1, 2, …, N) do
      for (i=1, 2, …, n) do
         if (b_{i,j,k} bids for service as_i) do
            update b_{i,j,k} into service cluster AS1_i
         end if
      end for
   end for
end for
for (i=1, 2, …, n) do
   for all AS1_{i,j} in AS1_i do
      if (AS1_{i,j}.score≥C_score) then
         update AS1_{i,j} into AS_i
      end if
   end for
end for
```

3.5.2 Winner determination algorithm

The winner determination algorithm (WDA) is used to determine which bid is the winner in each abstract service. Because the goal is to minimize the total virtual cost of service composition, the winner determination problem is expressed as 0–1 integer programming. To find the optimal service composition scheme, we use the PSO algorithm. The pseudocode is given in Algorithm 2.

3.5.3 PSO algorithm

In the proposed auction algorithm, the PSO algorithm (Kennedy and Eberhart, 1995), a global optimization algorithm based on swarm intelligence, is used to find the optimal solution. There have been many studies in this field, such as Li et al. (2015), Dong and Zhou (2017), and Zhang et al. (2019).

Algorithm 2 WDA(AS, m, W, WS)

Input: particle=∅
npop: number of particle sets
AS set
m_i: number of AS_i's. $m=\{m_1, m_2, \ldots, m_i\}$ (i=1: n)
W: weighted values of QoS attributes
WS←∅
Output: WS, $\text{WS}'_{i,j,-k}$
// find the winner set WS
WS←PSO(AS, m, WS, W)
for $b_{i,j,k}$∈WS **do**
  // find the winner set excluding $b_{i,j,k}$
  $\text{WS}'_{i,j,-k}$←PSO(AS, m, WS/$b_{i,j,k}$, W)
end for

In PSO, each particle has a velocity and a position, which are updated according to the flight experience:

$$v_i^{t+1}=\omega v_i^t+c_1r_1(\text{pbest}_i-x_i^t)+c_2r_2(\text{gbest}_i-x_i^t), \quad (9)$$

$$x_i^{t+1}=x_i^t+v_i^{t+1}, \quad (10)$$

where v_i^t and v_i^{t+1} are the velocity of the i^{th} particle in the t^{th} and $(t+1)^{\text{th}}$ iterations, respectively, ω is the inertia weight, c_1 and c_2 are the acceleration coefficients, r_1 and r_2 are random numbers in [0, 1], and pbest_i and gbest_i are the local and global best positions found by the i^{th} particle so far, respectively.

The pseudocode of the PSO algorithm is given in Algorithm 3.

Algorithm 3 PSO(AS, m, WS, W)

Input: population=∅
npop: number of particle sets
AS set
m_i: number of AS_i's. $m=\{m_1, m_2, \ldots, m_i\}$ (i=1:n)
maxt: maximum number of iterations
W: weighted values of QoS attributes
Output: Best solution
// initialization
for i=1, 2, …, npop **do**
  for j=1, 2, …, m **do**
    randomly generate Pop_i^j in a certain range
  end for
  f_i←fitness(Pop_i) using Eq. (7)
  if (f_i<f_i^l) **then**
    f_i^l←f_i
  end if
  f^g←f_i^l
end for
for i=1, 2, …, maxt **do**
  for j=1, 2, …, npop **do**
    update the velocity of Pop_j using Eq. (9)
    update the position of Pop_j using Eq. (10)
    for k=1, 2, …, m **do**
      if (Pop_j^k is out of the range) **then**
        reassign a new value to Pop_j^k
      end if
    end for
    f_j←fitness(Pop_j) using Eq. (7)
    if (f_j<f_j^l) **then**
      f_j^l←f_j
    end if
    if (f_j^l<f^g) **then**
      f^g←f_j^l
    end if
  end for
end for

3.5.4 Payment calculation algorithm

The payment calculation algorithm is used to calculate the price of each winning bid. The pseudocode is presented in Algorithm 4.

Algorithm 4 PaymentCalculation(WS, $\text{WS}'_{i,j,-k}$, W)

Input: WS
$\text{WS}'_{i,j,-k}$
m_i: number of AS_i's. $m=\{m_1, m_2, \ldots, m_i\}$ (i=1:n)
r: number of service providers
W: weighted values of QoS attributes
Output: Price
for $b_{i,j,k}$∈WS **do**
  for $b_{i',j',k}$∈$\text{WS}'_{i,j,-k}$ **do**
    calculate $\text{vc}_{i',j',k'}$ using Eq. (6)
    $C_{-k}=C_{-k}+\text{vc}_{i',j',k'}$
  end for
  for b_{i^*,j^*,k^*}∈WS **do**
    if (i≠j) **then**
      calculate vc_{i^*,j^*,k^*} using Eq. (6)
      $C_k=C_k+\text{vc}_{i^*,j^*,k^*}$
    end if
  end for
  $P_{i,j,k}=C_{-k}-C_k$
end for

3.5.5 Complete auction algorithm

The complete auction algorithm is shown in Algorithm 5. The service selection algorithm is used to select the bids that meet the requirements of the user's QoS constraint in line 10. Then the winner determination algorithm is used to find the winning bids in line 11, where Algorithms 2 and 3 are used to find the globally optimal service composition. Finally, the price for each winning bid is calculated by Algorithm 4 in line 12.

Algorithm 5 Complete auction

```
 1: Input: B
 2: N: number of B's
 3: r: number of service providers
 4: n: number of abstract services
 5: W: weighted values of QoS attributes
 6: AS1=∅
 7: AS=∅
 8: WS←∅
 9: maxt: maximum number of iterations
    // service selection
10: AS set←ServiceSelection(B, r, n)
    // winner determination
11: WS, WS'_{i,j,−k}←WDA(AS, m, W, WS)
    // payment calculation
12: Price←PaymentCalculation(WS, WS'_{i,j,−k}, W)
```

4 Experimental results

4.1 Experimental settings

We have conducted the experiments on a workstation with two Intel® Xeon® Gold 5120 CPUs and 128 GB memory.

The experimental data were adopted from a real QoS data set which was collected over the Internet (Al-Masri and Mahmoud, 2007), including 2500 cloud services. Based on the same statistical distribution principle, the real data set was extended with random numbers, and the final data set contained 5000 cloud services, including service attributes such as response time, availability, and reliability, but without the costs of services. Similar to many studies (Xu et al., 2015), in this study we generated the costs of services which conformed to a normal distribution in [1, 10].

The performances of the auction algorithm to be studied include mainly the following: whether it is strategy-proof, the success ratio of service composition, and the utilities of the user and service providers.

4.2 Truthfulness

We verified whether the SCAUP auction mechanism achieved truthfulness by evaluating how the utility of a service provider varied when reporting higher or lower cost of a service. We set v=1 for SCAUP and 0 for VCG, and W={0.4, 0.3, 0.3}. Fig. 2 shows the relationship between the utility of the winner in an abstract service and its reported cost in SCAUP and VCG.

According to the experimental results, the 15$^{\text{th}}$ service provider of abstract service as_1 won the bid with a reported cost of 5.7 in both auction mechanisms. Assuming that this reported cost was the true cost of the service provider, the payment obtained by the service provider was 6.4488 for SCAUP and 6.2 for VCG, and the utility was 0.7488 for SCAUP and 0.5 for VCG. When the service provider reported a false cost, either higher or lower, the utility kept the same while the service provider was still the winner, because its real cost and the payment remained unchanged. However, when the reported cost was no less than 6.2, the virtual cost of the service provider in the SCAUP mechanism was no longer the lowest. In this scenario, the service provider lost the auction and its utility was zero. In the VCG mechanism, the utility of the service provider dropped to zero when the reported cost was no less than 6.3, indicating that the additional cost of the original winner was lower than 0.2488 which ranked two.

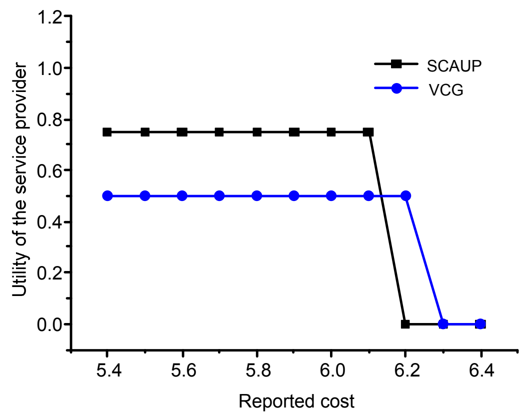

Fig. 2 Utility of the service provider in an abstract service varying with the reported cost

4.3 Individual rationality

In Figs. 3 and 4, we evaluated whether the SCAUP auction mechanism achieved individual rationality; i.e., the utilities of service providers and the user were no less than zero. We considered 50 abstract services; each abstract service had 100 candidate concrete services, v=1, the maximum response time of services was 2250 ms, and the minimum reliability and availability were both 60%. As can be observed from Fig. 3, the utilities of all service providers were greater than zero.

Fig. 4 shows the utility of the user obtained from 100 iterations. We set the maximum acceptable total price for the user as 450. As can be seen, the utility of the user was greater than zero in each calculation, almost equal to 121.2733, indicating that our proposed algorithm can find the optimal solution with good stability.

According to Figs. 3 and 4, the utilities of both the user and service providers were greater than zero, indicating that the proposed auction mechanism achieved individual rationality.

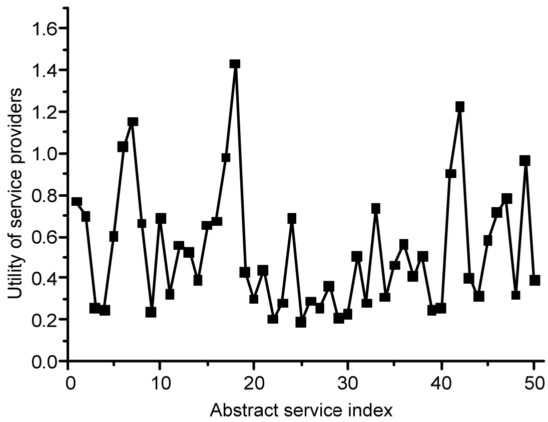

Fig. 3 Utility of winning service providers in each abstract service

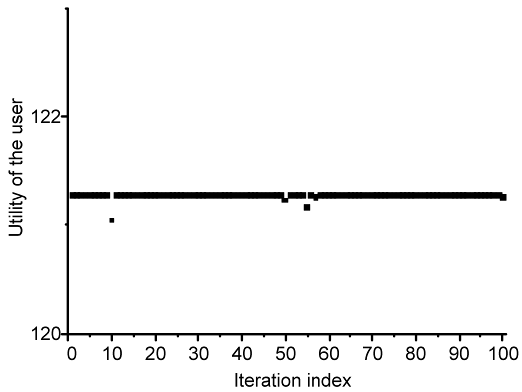

Fig. 4 Utility of the user in each repeated calculation

4.4 QoS growth rate

The performance of the SCAUP auction mechanism in improving the quality of the winner's service compared to the VCG mechanism was studied in this subsection. Fig. 5 shows the variation of the average QoS growth rate of the winner's abstract service with the increase of additional cost adjustment factor v.

There were 50 abstract services in the experiment and each abstract service had 100 services. Other parameters were set as follows: W={0.4, 0.3, 0.3}, the maximum response time was 2250 ms, the minimum reliability and availability were both 60%, and P=12×(number of abstract services).

As can be seen from Fig. 5, with v increasing from 1 to 50, the levels of all three QoS attributes were improved. The response time had the most obvious improvement, increasing from 34% to 143% compared with the VCG result, the reliability can be improved by 17%, and the availability can be improved by 4%. The growth rates for availability and reliability were relatively stable when v>25, and the growth rate in response time was relatively stable when v>10 and increased again when v>30. The response time improved a lot, which is related to the large distribution range and fluctuation of the data.

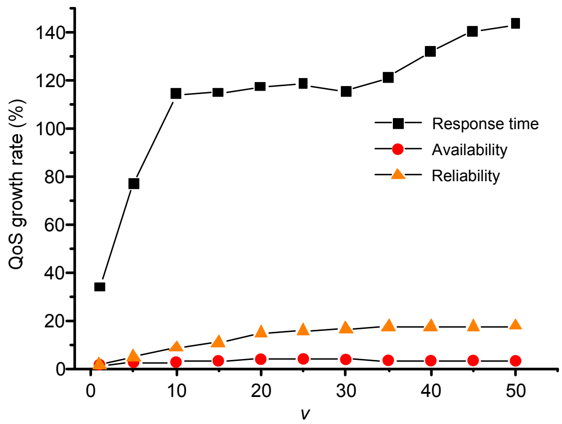

Fig. 5 QoS growth rate based on adjustment factor *v*

The above phenomenon showed that the proposed auction mechanism can effectively improve the QoS level of service composition and enhance the quality of services that the user receives. However, note that when v>25, the total price of service composition was beyond the maximum price acceptable to the user. Therefore, in this example, a v between 10 and 20 was a good choice for the user. In other cases, the user can choose weight W and additional cost adjustment factor v by comprehensively considering his/her demand and payment ability.

4.5 Social welfare

In this subsection, we evaluated the user's social welfare performance from the perspectives of the user preference weight, additional cost adjustment factor, number of bids, and QoS attributes. All the experimental results were averaged over 50 independent calculations.

4.5.1 User preference weight

Assume that the user's weight preferences were $W=\{\omega_1, \omega_2, \omega_3\}$, where ω_1, ω_2, and ω_3 represent the weights of response time, availability, and reliability, respectively. Because $\omega_1+\omega_2+\omega_3=1$, we need only to study the influence of ω_1 and ω_2 on social welfare. In this study, ω_1 and ω_2 took random positive real values with their sum not greater than 1. Based on this premise, 250 Monte-Carlo random calculations were performed, and the social welfare results are shown in Fig. 6. Other parameters were set as follows: v=1, the maximum response time was 2250 ms, and the minimum reliability and availability were both 60%. There were 50 abstract services in total, with 100 service bids under each abstract service.

Fig. 6 shows that the change of the user preference weight had little impact on the user's social welfare. The smaller ω_1 and ω_2, the greater the social welfare. When ω_1 had a larger value, less social welfare was achieved. Fig. 6 can be summarized as follows: the higher the reliability weight, the greater the social welfare; the higher the response time weight, the less the social welfare. In addition, the data points formed a plane in the three-dimensional space, indicating that the influence of each weight was linear.

4.5.2 Additional cost adjustment factor

Fig. 7 shows the impact of changes in the additional cost adjustment factor v on social welfare. The data points corresponding to six types of situations with different numbers of abstract services were presented, where each abstract service had 100 services. Other parameters were set as follows: W={0.4, 0.3, 0.3}, the maximum response time was 2250 ms, and the minimum reliability and availability were both 60%.

As can be seen from Fig. 7, when the number of abstract services was fixed, the social welfare (total price) increased with the increase of v. This is because when v increases, the user has a stronger demand for high-quality services. At this point, the additional cost calculated by QoS increases the virtual cost of each bidding service, thus increasing the total price.

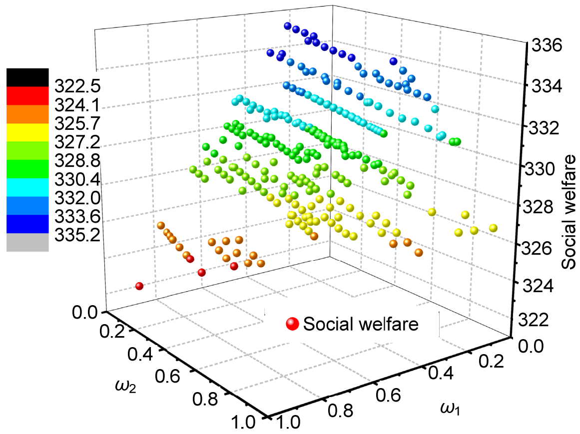

Fig. 6 Social welfare varying with ω_1 and ω_2

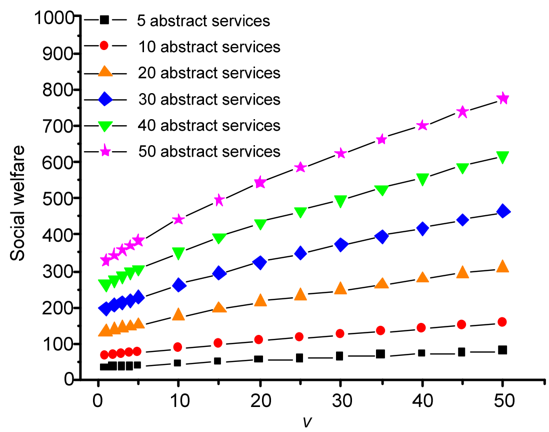

Fig. 7 Social welfare varying with adjustment factor v

When v was a constant, as the number of abstract services increased, social welfare increased. As shown in Fig. 7, the larger the number of abstract services, the higher the position of the data points. The reason is obvious, as the user demands more services.

However, note that the maximum total price P that the user can afford is limited. If the user cannot afford P, the auction fails. On the other hand, in our auction mechanism, the user preference for QoS is reflected in the form of additional cost; that is, the user's right to choose QoS is at the cost of paying more. Therefore, the user needs to consider his/her own needs and payment capacity to set W and v in the auction, so as to balance cost and benefit.

4.5.3 Impact of the number of bids

In this subsection, the influence of the number of bids on social welfare under each abstract service in SCAUP and VCG mechanisms was studied (Fig. 8). The number of abstract services was fixed at 50, and the number of concrete service bids under each abstract service varied from 10 to 100. For the VCG mechanism, other calculation parameters were set as follows: the maximum response time was 2250 ms, and the minimum reliability and availability were 60%. For the SCAUP mechanism, the additional parameters were v=1 and W={0.4, 0.3, 0.3}.

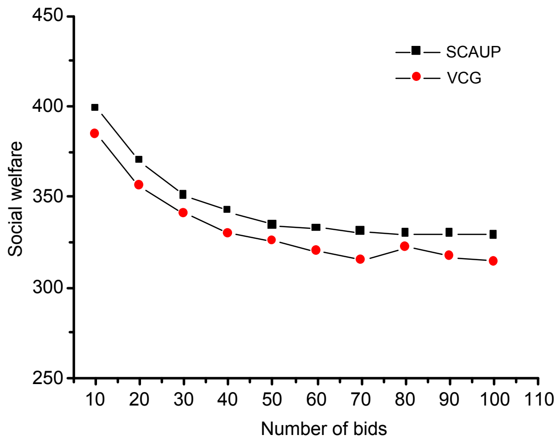

Fig. 8 Social welfare varying with the number of bids

As can be seen from Fig. 8, as the number of bids increased, social welfare (i.e., total price) of SCAUP and VCG showed a declining trend during the whole auction process. This is because as the number of bids increases, competition among sellers (i.e., service providers) becomes more intense, and new services with lower prices and better quality may join the bidding process, thus reducing the total price of the service composition scheme.

However, the social welfare of the VCG auction mechanism was generally slightly lower than that of the SCAUP auction mechanism. This is because when v>0, the additional cost increases the virtual costs of the services, which is the cost of pursuing a high-level QoS.

4.5.4 Impact of QoS attributes

In this subsection we studied the impact of SCAUP and VCG auction mechanisms on the social welfare when the user changed its QoS constraints. There were three QoS attributes: response time, availability, and reliability. Two of them were fixed and the other one was changed. Since Fig. 7 has shown the rule of generality, the influence of the abstract service number was not discussed here. The number of abstract services was fixed at 50. The other parameters were set as follows: v=1, W={0.4, 0.3, 0.3}, there were 100 service bids under each abstract service, and the minimum reliability and availability were fixed at 60%.

Fig. 9 shows the change in social welfare varying with the response time; the maximum response time changed from 2250 to 50 ms. As can be seen, the social welfare in SCAUP and VCG auction mechanisms increased as the response time constraint of the service was tightened from 2250 to 100 ms. The reason is that when the response time constraint is tightened, the number of services available is reduced and some winners are excluded, thus causing the social welfare to rise. When the response time constraint was tightened to 50 ms, the auction failed as no bid can satisfy these constraints, and social welfare was reduced to zero.

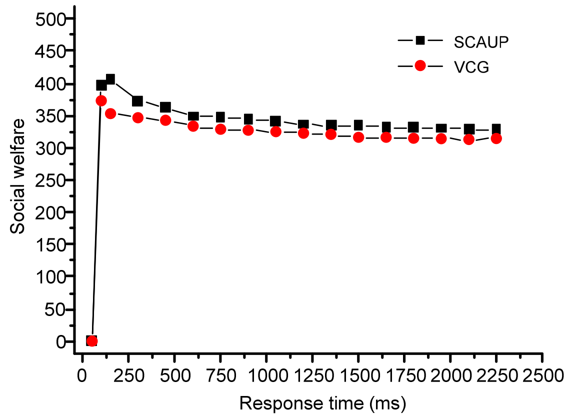

Fig. 9 Social welfare varying with the response time

Similarly, Figs. 10 and 11 show the changes of social welfare varying with the availability and reliability, respectively. The minimum availability and reliability constraints were tightened from 5% to 100%, and the maximum response time was 2250 ms. When the availability constraint changed, the reliability was fixed at 60%, and vice versa. Both figures show an upward trend as the constraint was tightened, but when the availability was 100% and the reliability 85%, the auction failed, and the social welfare dropped to zero. The reason is similar to that for the response time.

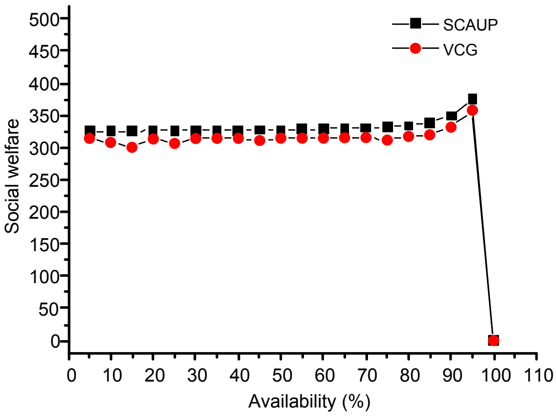

Fig. 10 Social welfare varying with the availability

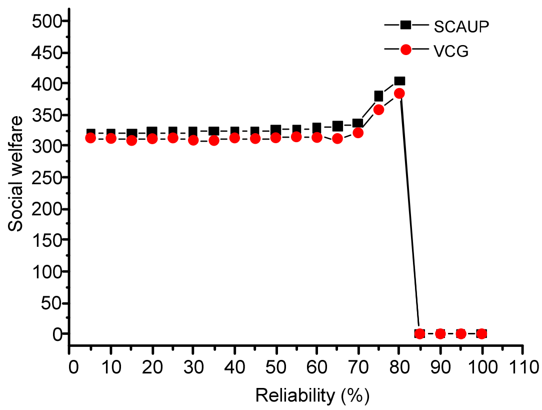

Fig. 11 Social welfare varying with the reliability

Figs. 9–11 show that the social welfare of the SCAUP mechanism was slightly larger than that of the VCG mechanism. This does not mean that VCG is superior to SCAUP. Note that the design intention of the SCAUP mechanism is to support the user in selecting and composing services based on his/her preferences for QoS attributes and to meet the strategy-proof property.

4.6 Success ratio

In this subsection, we studied the influence of different QoS constraints and the number of abstract services on the auction success ratio. The success ratio refers to the proportion of the total number of successful repetitions (n_s) to the total number of independent repetitions (N) of the auction algorithm:

$$\text{Success ratio} = n_s / N \times 100\%. \tag{11}$$

Successful service composition occurs when the following situations do not occur: (1) After excluding the bids that do not meet QoS constraints, there is at least one abstract service having no concrete service to choose; (2) The total price exceeds the maximum price P that the user can afford, and P=12×(total number of abstract services); (3) The price of at least one abstract service is less than the winner's reported cost. All data points were the results of 50 independent calculations under each calculation condition.

Fig. 12 shows that the success ratio was around 100% in most cases, but was reduced to zero when the response time was 50 ms.

Fig. 13 shows that the success ratio was 100%, but was reduced to zero when the availability was 100%. Fig. 14 shows a similar trend, where the success ratio was zero when the reliability was greater than 70%.

The above indicates that the numbers of abstract services and QoS attribute constraints have less influence on the success ratio. So, our proposed auction algorithm has better comprehensive performance and higher stability.

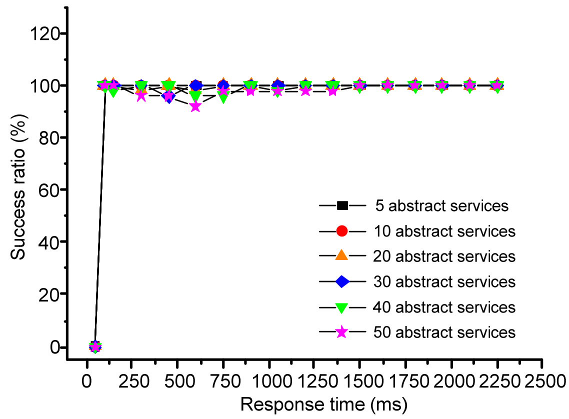

Fig. 12 Success ratio varying with the response time

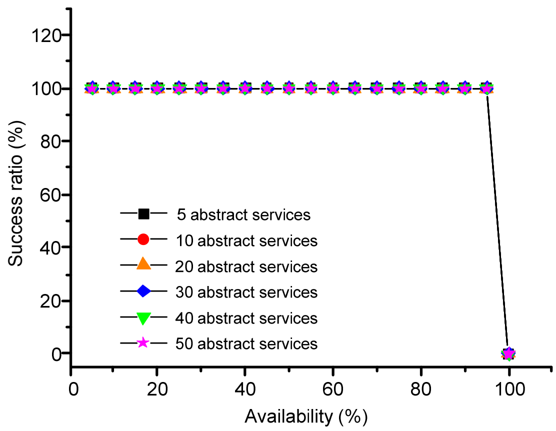

Fig. 13 Success ratio varying with the availability

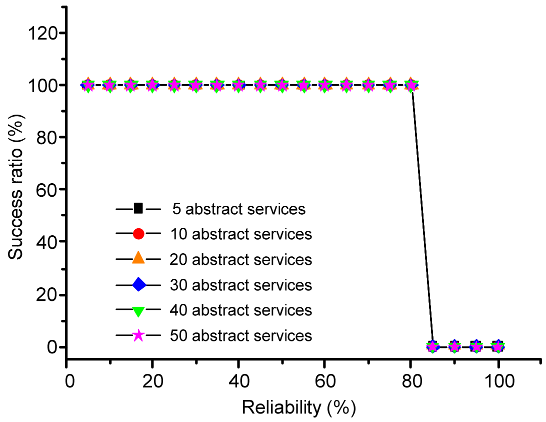

Fig. 14 Success ratio varying with the reliability

5 Related works

Due to the development of computer and network technology, users have a large number of services to choose to compose the complex business workflows they need. Cloud computing is one of the main applications. For example, Ghahramani et al. (2017) analyzed the QoS architecture for cloud computing. However, the existing price model for purchasing service is a fixed pricing model set by service providers, which is very disadvantageous to users. At present, auction has become a popular dynamic pricing mechanism that is widely used in the field of cloud computing, where user preference is a difficult aspect to consider. This section is focused on the application of auctions in cloud computing for service composition or resource allocation issues, which is summarized into two categories: studies that consider or do not consider user preferences.

5.1 Studies not considering user preferences

There are many studies on the service composition problem that do not involve user preferences. He et al. (2014) proposed a novel approach called the combinatorial auction for service selection. They intended to implement efficient service selection in service-based systems. In the auction, service providers can bid for combinations of services and provide discounts. Wei et al. (2016) studied an infrastructure as a service (IaaS) market's competitive and bidding scenario based on the auction mechanism for independent users. They proposed a distributed bidding adjustment approach based on feedback. Each user can find approximate optimal bids and the resulting allocation of resources benefits all parties. Moghaddam and Davis (2017) studied service selection approaches based on auction mechanisms, and proposed a design framework that introduces the elements in an auction-based model. Dimitriou and Krontiris (2017) studied the security and privacy issues in mobile crowdfunding and presented a multi-attribute reverse auction mechanism to motivate service providers to submit data of better quality. The mechanism offers strong guarantees of security and privacy. Borjigin et al. (2018) developed a double-auction approach for both service function chain routing and price adjustment of network function virtualization. Shi et al. (2017) assumed that there is a double auction marketplace managed by the cloud platform. Based on the game theory, they analyzed the Bayes-Nash equilibrium bidding strategies for the service.

The above studies use an auction mechanism in cloud computing to realize various complex functional requirements, but these auction mechanisms do not take QoS into account. There are some studies that consider certain QoS requirements for the process of service composition.

For example, Wang et al. (2017a) studied the QoS-aware web service composition problem, and proposed an incentive mechanism. The mechanism can choose the optimal web service and shows that each web service bids truthfully. Similarly, Wang et al. (2017b) proposed a multi-round Vickrey auction, and the results satisfied the global QoS constraint. Wang and Du (2019) proposed an incentive contract or mechanism that can motivate service providers to offer satisfactory prices and QoS. The solution that satisfies the global QoS requirements can be found through iteration. Although QoS has been considered in the above works, user preferences for QoS were not considered.

In many auction studies, the most important characteristic is strategy-proof, which is not easy to achieve. The VCG mechanism satisfies truthfulness and individual rationality. As mentioned in the introduction, there are many studies that apply VCG auctions to service composition problems. Tanaka and Murakami (2016) proposed a service selection algorithm based on dynamic programming and a VCG payment model. They extended the mechanism and improved performance through iteration of the service selection process. Similarly, Wu QW et al. (2018) proposed a VCG auction-based dynamic pricing mechanism to solve a generalized service composition. Service providers bid for services of different granularities and a user chooses a composition that can minimize the social cost and satisfy quality constraints. Jiang et al. (2018) studied the on-demand mechanism design of the IaaS, including resource allocation and pricing issues under dynamic scenarios. They designed an auction mechanism based on VCG which guaranteed incentive compatibility and individual rationality. Watanabe et al. (2012) proposed an algorithm based on a Vickrey auction to solve the service selection problem. The service selection process was divided into two stages, avoiding full exploration of service composition and improving the efficiency of computation while encouraging providers to offer their best QoS.

On the other hand, there are studies that do not use the VCG mechanism to satisfy the strategy-proof property. Prasad and Rao (2014) proposed a cloud resource procurement approach that implements dynamic pricing. They proposed three possible mechanisms and only the cloud optimal mechanism (C-OPT) guaranteed both Bayesian incentive compatibility and individual rationality. Zhou et al. (2019) studied the offloading market in a hybrid/heterogeneous mobile cloud, and proposed a reverse real-time auction mechanism. The proposed auction algorithm demonstrated truthfulness and individual rationality. Although the above studies on service composition auction or service selection problems have the strategy-proof property, the participants' preferences for QoS are not considered.

5.2 Studies on user preferences in auctions

Considering user preferences in the auction mechanism under a cloud computing environment is a challenging research direction, and there have been some studies. Based on combinatorial auctions, Moghaddam et al. (2013) proposed a novel approach to implement dynamic service composition. The approach allows incorporating preferences from service providers and users. Moghaddam (2012) proposed a combinatorial procurement auction mechanism to solve service composition problems. The mechanism supports dynamic pricing and enables providers to express their preferences. The providers are then encouraged to offer truthful prices. Karakaya and Köksalan (2011) proposed an interactive approach in a multi-attribute, single-item, multi-round, reverse auction environment, estimating the possible future preferences of the buyer using past preferences.

Although the above studies applied auction mechanism to the service composition problem and considered the preferences of participants, no systematic studies have been conducted to satisfy the strategy-proof property of the auction mechanism and to take account of participant preferences in payment calculation.

There have been some studies that involve cloud workflow scheduling, web service discovery, and web service composition, such as Wang et al. (2014, 2016), Deng et al. (2016), Wu Y et al. (2016), Sha et al. (2019), and Wu QW et al. (2020).

In summary, there is currently no literature that implements such an auction based on the reverse VCG mechanism to deal with the service composition problem: taking into account user preferences for QoS in both winner determination and payment calculation, and meeting the strategy-proof property.

6 Conclusions

In this paper, we have studied user preferences in service composition auction and proposed a new service auction mechanism based on user preferences (SCAUP). SCAUP is based on VCG composition reverse auction; the service preferred by the user is selected to meet the user's diversified demands for different QoS attributes. In this mechanism, new winner determination and payment calculation methods were proposed which allow the user to evaluate a service according to the importance to each QoS attribute. It was also proved theoretically that the SCAUP auction mechanism has the strategy-proof property. We conducted experiments based on real data and verified that the SCAUP auction mechanism has the strategy-proof property and can effectively improve the overall QoS level of service composition. The effects of the calculated parameters on the social welfare and success ratio have been systematically evaluated. Experimental results showed that when the additional cost adjustment factor v was increased, the user's QoS was improved and the payment increased. Therefore, the user needs to take his/her own needs and payment capacity into consideration when choosing user preferences W and v in the auction, to balance cost and benefit.

In the future we will apply our proposed SCAUP auction mechanism to new frontier fields and more complex systems, and optimize the proposed auction algorithm to improve the computational efficiency and performance.

Contributors

Yao XIA and Zhiqiu HUANG designed the research. Yao XIA processed the data and drafted the manuscript. Zhiqiu HUANG helped organize the manuscript. Yao XIA revised and finalized the paper.

Compliance with ethics guidelines

Yao XIA and Zhiqiu HUANG declare that they have no conflict of interest.

References

Al-Masri E, Mahmoud QH, 2007. Discovering the best web service. Proc 16th Int Conf on World Wide Web, p.1257-1258. https://doi.org/10.1145/1242572.1242795

Borjigin W, Ota K, Dong MX, 2018. In broker we trust: a double-auction approach for resource allocation in NFV markets. *IEEE Trans Netw Serv Manag*, 15(4):1322-1333. https://doi.org/10.1109/TNSM.2018.2882535

Deng SG, Huang LT, Taheri J, et al., 2016. Mobility-aware service composition in mobile communities. *IEEE Trans Syst Man Cybern Syst*, 47(3):555-568. https://doi.org/10.1109/TSMC.2016.2521736

Dimitriou T, Krontiris I, 2017. Privacy-respecting auctions and rewarding mechanisms in mobile crowd-sensing applications. *J Netw Comput Appl*, 100:24-34. https://doi.org/10.1016/j.jnca.2017.10.012

Dong WY, Zhou MC, 2017. A supervised learning and control method to improve particle swarm optimization algorithms. *IEEE Trans Syst Man Cybern Syst*, 47(7):1135-1148. https://doi.org/10.1109/TSMC.2016.2560128

Ghahramani MH, Zhou MC, Hon CT, 2017. Toward cloud computing QoS architecture: analysis of cloud systems and cloud services. *IEEE/CAA J Autom Sin*, 4(1):6-18. https://doi.org/10.1109/JAS.2017.7510313

He Q, Yan J, Jin H, et al., 2014. Quality-aware service selection for service-based systems based on iterative multi-attribute combinatorial auction. *IEEE Trans Softw Eng*, 40(2):192-215. https://doi.org/10.1109/TSE.2013.2297911

Jiang CX, Chen Y, Wang Q, et al., 2018. Data-driven auction mechanism design in IaaS cloud computing. *IEEE Trans Serv Comput*, 11(5):743-756. https://doi.org/10.1109/TSC.2015.2464810

Karakaya G, Köksalan M, 2011. An interactive approach for multi-attribute auctions. *Dec Support Syst*, 51(2):299-306. https://doi.org/10.1016/j.dss.2010.11.023

Kennedy J, Eberhart R, 1995. Particle swarm optimization. Proc IEEE Int Conf on Neural Networks, p.1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Li J, Zhang JQ, Jiang CJ, et al., 2015. Composite particle swarm optimizer with historical memory for function optimization. *IEEE Trans Cybern*, 45(10):2350-2363. https://doi.org/10.1109/TCYB.2015.2424836

Moghaddam M, 2012. An auction-based approach for composite web service selection. Int Workshops on Service Oriented Computing, p.400-405. https://doi.org/10.1007/978-3-642-37804-1_41

Moghaddam M, Davis JG, 2017. Auction-based models for composite service selection: a design framework. Proc 5th and 6th Australian Symp on Service Research and Innovation, p.101-115. https://doi.org/10.1007/978-3-319-76587-7_7

Moghaddam M, Davis JG, Viglas T, 2013. A combinatorial auction model for composite service selection based on preferences and constraints. IEEE Int Conf on Services Computing, p.81-88. https://doi.org/10.1109/SCC.2013.112

Mu'Alem A, Nisan N, 2008. Truthful approximation mechanisms for restricted combinatorial auctions. *Game Econ Behav*, 64(2):612-631. https://doi.org/10.1016/j.geb.2007.12.009

Nisan N, Roughgarden T, Tardos E, et al., 2007. Algorithmic Game Theory. Cambridge University Press, Cambridge, USA.

Prasad AS, Rao S, 2014. A mechanism design approach to resource procurement in cloud computing. *IEEE Trans Comput*, 63(1):17-30. https://doi.org/10.1109/TC.2013.106

Sha J, Du YY, Qi L, 2019. A user requirement oriented web service discovery approach based on logic and threshold Petri net. *IEEE/CAA J Autom Sin*, 6(6):1528-1542. https://doi.org/10.1109/JAS.2019.1911657

Shi B, Wang JW, Wang ZW, et al., 2017. Trading web services in a double auction-based cloud platform: a game theoretic analysis. Proc IEEE 14th Int Conf on Services Computing, p.76-83. https://doi.org/10.1109/SCC.2017.18

Singer Y, 2010. Budget feasible mechanisms. Proc 51st Annual Symp on Foundations of Computer Science, p.765-774. https://doi.org/10.1109/FOCS.2010.78

Tanaka M, Murakami Y, 2016. Strategy-proof pricing for cloud service composition. *IEEE Trans Cloud Comput*, 4(3):363-375. https://doi.org/10.1109/TCC.2014.2338310

Wang PW, Du XY, 2019. QoS-aware service selection using an incentive mechanism. *IEEE Trans Serv Comput*, 12(2):262-275. https://doi.org/10.1109/TSC.2016.2602203

Wang PW, Ding ZJ, Jiang CJ, et al., 2014. Constraint-aware approach to web service composition. *IEEE Trans Syst Man Cybern Syst*, 44(6):770-784.

https://doi.org/10.1109/TSMC.2013.2280559

Wang PW, Ding ZJ, Jiang CJ, et al., 2016. Automatic web service composition based on uncertainty execution effects. *IEEE Trans Serv Comput*, 9(4):551-565. https://doi.org/10.1109/TSC.2015.2412943

Wang PW, Liu T, Zhan Y, et al., 2017a. A Bayesian Nash equilibrium of QoS-aware web service composition. IEEE Int Conf on Web Services, p.676-683. https://doi.org/10.1109/ICWS.2017.81

Wang PW, Zhan Y, Liu T, et al., 2017b. QoS-aware service composition for service-based systems using multi-round vickery auction. IEEE Int Conf on Systems, Man, and Cybernetics, p.2891-2896. https://doi.org/10.1109/SMC.2017.8123066

Watanabe A, Ishikawa F, Fukazawa Y, et al., 2012. Web service selection algorithm using Vickrey auction. Proc IEEE 19th Int Conf on Web Services, p.336-342. https://doi.org/10.1109/ICWS.2012.83

Wei Y, Pan L, Yuan D, et al., 2016. A distributed game-theoretic approach for IaaS service trading in an auction-based cloud market. IEEE TrustCom/BigDataSE/ISPA, p.1543-1550. https://doi.org/10.1109/TrustCom.2016.0240

Wen YT, Shi JY, Zhang Q, et al., 2015. Quality-driven auction-based incentive mechanism for mobile crowd sensing. *IEEE Trans Veh Technol*, 64(9):4203-4214. https://doi.org/10.1109/TVT.2014.2363842

Wu QW, Zhou MC, Zhu QS, et al., 2018. VCG auction-based dynamic pricing for multigranularity service composition. *IEEE Trans Autom Sci Eng*, 15(2):796-805. https://doi.org/10.1109/TASE.2017.2695123

Wu QW, Zhou MC, Zhu QS, et al., 2020. MOELS: multi-objective evolutionary list scheduling for cloud workflows. *IEEE Trans Autom Sci Eng*, 17(1):166-176. https://doi.org/10.1109/TASE.2019.2918691

Wu Y, Yan CG, Ding ZJ, et al., 2016. A multilevel index model to expedite web service discovery and composition in large-scale service repositories. *IEEE Trans Serv Comput*, 9(3):330-342. https://doi.org/10.1109/TSC.2015.2398442

Xu J, Xiang JX, Yang DJ, 2015. Incentive mechanisms for time window dependent tasks in mobile crowdsensing. *IEEE Trans Wirel Commun*, 14(11):6353-6364. https://doi.org/10.1109/TWC.2015.2452923

Zhang Y, Zhou P, Cui GM, 2019. Multi-model based PSO method for burden distribution matrix optimization with expected burden distribution output behaviors. *IEEE/CAA J Autom Sin*, 6(6):1506-1512. https://doi.org/10.1109/JAS.2018.7511090

Zheng ZZ, Gui Y, Wu F, et al., 2015. STAR: strategy-proof double auctions for multi-cloud, multi-tenant bandwidth reservation. *IEEE Trans Comput*, 64(7):2071-2083. https://doi.org/10.1109/TC.2014.2346204

Zheng ZZ, Wu F, Gao XF, et al., 2017. A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing. *IEEE Trans Mob Comput*, 16(9): 2392-2407. https://doi.org/10.1109/TMC.2016.2632721

Zhou BW, Srirama SN, Buyya R, 2019. An auction-based incentive mechanism for heterogeneous mobile clouds. *J Syst Softw*, 152:151-164. https://doi.org/10.1016/j.jss.2019.03.003