Ann. Telecommun.
DOI 10.1007/s12243-010-0184-0

A market-oriented dynamic collaborative cloud services platform

Mohammad Mehedi Hassan · Biao Song · Eui-Nam Huh

Received: 31 May 2009 / Accepted: 28 May 2010

Abstract Currently, interoperability and scalability are two major challenging issues for cloud computing. Forming a dynamic collaboration (DC) platform among cloud providers (CPs) can help to better address these issues. A DC platform can facilitate expense reduction, avoiding adverse business impacts and offering collaborative or portable cloud services to consumers. However, there are two major challenges involved in this undertaking; one is to find an appropriate market model to enable a DC platform, and the other one is to minimize conflicts among CPs that may occur in a market-oriented DC platform. In this paper, we present a novel combinatorial auction (CA)-based cloud market (CACM) model that enables a DC platform in CPs. To minimize conflicts among CPs, a new auction policy is proposed that allows a CP to dynamically collaborate with suitable partner CPs to form groups and publishes their group bids as a single bid to compete in the auction. However, identifying a suitable combination of CP partners to form the group and reduce conflicts is a NP-hard problem. Hence, we propose a promising multi-objective (MO) optimization model for partner selection using individual information and past collaborative relationship information, which is seldom considered. A multi-objective genetic algorithm (MOGA) called MOGA-IC is proposed to solve the MO optimization problem. This algorithm is developed using two popular MOGAs, the non-dominated sorting genetic algorithm (NSGA-II) and the strength pareto evolutionary genetic algorithm (SPEA2). The experimental results show that MOGA-IC with NSGA-II outperformed the MOGA-IC with SPEA2 in identifying useful pareto-optimal solution sets. Other simulation experiments were conducted to verify the effectiveness of the MOGA-IC in terms of satisfactory partner selection and conflict minimization in the CACM model. In addition, the performance of the CACM model was compared to the existing CA model in terms of economic efficiency.

M. M. Hassan · B. Song · E.-N. Huh (✉)
Internet Computing and Network Security Lab,
Department of Computer Engineering,
Kyung Hee University, Global Campus,
1 Seocheon-dong, Giheung-gu,
Yongin-si, Gyeonggi-do 446-701, South Korea
e-mail: johnhuh@khu.ac.kr

M. M. Hassan
e-mail: hassan@khu.ac.kr

B. Song
e-mail: bsong@khu.ac.kr

Keywords Cloud market · Combinatorial auction · Dynamic collaboration · Interoperability · Partner selection · Multi-objective genetic algorithm

1 Introduction

Emerging cloud computing [1] offers a new computing model in which resources such as computing power, storage, online applications, and networking infrastructures can be shared as "services" over the internet. Cloud providers are motivated by the profits to be made by charging consumers for accessing these services. Consumers, such as enterprises, are attracted by the opportunity for reducing or eliminating costs associated with "in-house" provisions of these services. However, existing commercial cloud services are proprietary in nature. They are owned and operated by individual companies (public or private), each of which has created its own closed network, which is expensive to establish and maintain. Running a global cloud service is even more costly, requiring an enormous amount of capital and labor.

In addition, consumers are restricted to offerings from a single provider at a time and hence cannot simultaneously use multiple or collaborative cloud services. The rapid take-up of some services, particularly those for infrastructure, requires portability between multiple cloud infrastructures. Thus, interoperability is becoming an important issue for cloud services since many enterprises do not want to tie their most important applications to specific providers' remote infrastructure or platforms [2]. For example, Salesforce.com Inc's Force.com platform now enables developers to use its cloud application development platform alongside Amazon Web Services LLC's infrastructure and storage services. To make cloud computing truly scalable, one huge cloud that is controlled by one huge vendor running a very narrow set of applications is not feasible.

Commercial cloud providers (CPs) make specific commitments to their customers by signing service level agreements (SLAs) [3], a contract between the service provider and the customer to describe the provider's commitment and to specify penalties if those commitments are not met. For example, cloud “bursting” (using remote resources to handle peaks in demand for an application) may result in an SLA violation and end up incurring additional costs for the provider.

To reduce operation costs, prevent adverse business effects, and offer collaborative or portable cloud services, there is a need for a dynamic collaboration (DC) [4] platform among CPs. In a DC platform, (1) each CP can share its own local resources/services with other partner CPs and so can access much larger pools of resources/services; (2) each provider can maximize its profits by offering existing service capabilities to collaborative partners so that they may create a new value-added collaborative service by mashing-up existing services, and thus interoperability issues can be resolved; (3) the peak-load handling capacity of every CP increases without maintaining or administering any additional computing nodes, services, or storage devices; and (4) the reliability of a CP is enhanced as a result of multiple redundant clouds that can efficiently tackle a disaster condition, ensuring business continuity.

However, there are two major challenges involved in creating such a DC platform. The first is to create and commercialize an appropriate cloud market model that can enable dynamic collaboration of cloud capabilities, hiring resources, and assembling new services. Such a market can provide opportunities to consumers as well as service-sharing incentives for CPs.

The second challenge is to minimize the large number of conflicts that may occur in a market-oriented DC platform when negotiating among providers. One reason for the occurrence of the large number of conflicts is that each provider must agree with the resources/services contributed by other providers against a set of its own policies in DC [8, 9]. Another reason is the inclusion of high collaboration costs (e.g., network establishment, information transmission, capital flow) by the providers with their bidding prices as they do not know with whom they need to collaborate after winning an auction.

In this paper, we discuss these two major challenges for forming a DC platform among CPs and present possible solutions. The main contributions of this paper are as follows:

– A novel combinatorial auction (CA)-based cloud market model with a new auction policy called CACM is proposed to facilitate a virtual organization (VO)-based dynamic collaboration platform among CPs. To the best of our knowledge, this is the first paper reporting on the formation of a market-oriented DC platform model among CPs.
– To address the issue of conflict minimization among providers, the existing auction policy of CA [5–7] is modified. The new auction policy in the CACM model allows a CP to dynamically collaborate with suitable partner CPs to form a group before joining the auction and to publish their group bids as a single bid to completely fulfill the service requirements, along with other CPs, who publish separate bids to partially fulfill the service requirements. This new approach can create more opportunities to win auctions since collaboration cost, negotiation time, and conflicts among CPs can be minimized.
– To find a good combination of CP partners for forming groups and reducing conflicts, a multi-objective (MO) optimization model for quantitatively evaluating the partners using their individual information (INI) and past collaborative relationship information (PRI) [27] is proposed. In the existing approaches [10–26] for partner selection, the INI is mostly used, while the PRI of partners is typically overlooked.
– To solve the MO optimization model for partner selection, MOGA-IC, a multi-objective genetic algorithm (MOGA) that uses INI and PRI, is also presented. We develop MOGA-IC using the two popular MOGAs, non-dominated sorting genetic algorithm (NSGA-II) [28], and strength pareto evolutionary genetic algorithm (SPEA2) [29], to find an appropriate diversity preservation mechanism for selecting operators to enhance the yield of pareto-optimal solutions during optimization with multiple conflicting objectives.
– We implement the proposed CACM model in a simulated environment and study its economic efficiency with the existing CA model. Moreover, a numerical example is presented to illustrate the proposed MOGA-IC with NSGA-II and SPEA2.
– In addition, we develop MOGA-I (multi-objective genetic algorithm using individual information), an existing partner selection algorithm, to validate the performance of MOGA-IC in the CACM model. Simulation experiments are conducted to show the effectiveness of the proposed

MOGA-IC compared to that of MOGA-I in terms of satisfactory partner selection and conflict minimization.

The rest of this paper is organized as follows: Section 2 describes related works in the literature. Section 3 shows a DC platform formed among CPs. Section 4 describes the proposed CACM model and system model for auction. In Section 5, we present the model of CP partner selection and the proposed MOGA-IC. In Section 6, simulation results are presented to show the effectivenesses of the CACM model and MOGA-IC. We conclude our work by presenting a summary and describing future works in Section 7.

2 Related work

Cloud computing evolved rapidly during 2008, and it is a current hot topic for research. However, no work has been found in the literature regarding the establishment of a dynamic collaboration platform among CPs. There are a few approaches proposed in the literature regarding the cloud market model. In [3], the authors present a vision of the twenty-first century computing, describe some representative platforms for cloud computing covering the state-of-the-art, and provide the architecture for creating a general auction-based cloud market for trading cloud services and resource management. However, this market model cannot be directly applicable to the creation of a DC platform among CPs since the DC platform deals with a combinatorial allocation problem.

There are three types of auctions, one-sided (e.g., first price and Vickrey auctions), double-sided (e.g., double auction), and combinatorial (CA) [5–7, 30, 31]. To enable the DC platform among CPs, CA is the appropriate market mechanism. In the CA-based market model, the user/consumer can bid a price value for a combination of services, instead of bidding separately for each task or service, and each bidder or service provider is allowed to compete for a set of services.

However, the existing auction policy of the CA-based market model is not fully capable of meeting the requirements of a DC platform. If the existing auction policy of the CA model is applied, each bidder (CP) is allowed to separately compete for a set of services. After the bidding, the winning bidders need to collaborate with each other. As we mentioned earlier, a large number of conflicts may occur when negotiating among providers in the DC platform [8, 9]. The CA-based market model cannot address the issue of conflict minimization among the CPs in a DC platform.

The current approaches to handling conflicts are to design eContract delivery sequences [8, 9]. An eContract [32] is used to capture the contributions as well as the agreements among all participants. However, the main problem of these approaches is that an auctioneer may choose an improper set of service providers (competing or rival companies). Then, despite the arrangement of the delivery service of the eContract, a large number of conflicts cannot be prevented from occurring. We propose to modify the existing auction policy of CA that allows the CPs to publish their bids collaboratively into a single bid in the auction by dynamically collaborating with suitable partners. This approach can help to minimize conflicts and collaboration costs among CPs since they often know each other very well. This technique will also increase the group's chances of winning the auction.

Nevertheless, the collaborator or partner selection problem (PSP) is a complex problem which usually involves a large quantity of factors (quantitative or qualitative ones) and has been proven to be NP-hard [10] or NP-complete [11]. For CP partner selection, for instance, cost and quality of service are the most important factors. Furthermore, PSP for CPs in the CACM model is different from other PSP problems in areas like manufacturing, supply chain, or virtual enterprise [10–26] since a large number of conflicts may occur among CPs due to dynamic collaboration. In the existing studies on partner selection, the INI is most commonly used, while the PRI [27] between partners is overlooked. In fact, the success of past relations between participating CPs may reduce uncertainty and conflicts, shorten the adaptation duration, and help with performance promotion. Existing methods cannot be applied directly to solve the PSP problem of CPs. Therefore, an appropriate MO optimization model using INI, PRI, and an effective MOGA called MOGA-IC to solve the MO optimization problem is proposed. None of the MOGAs available in the literature [12, 14–17, 26] consider PRI for partner selection.

3 Dynamic collaborative cloud services platform

Dynamic collaboration is a viable business model in which each participant shares their own local resources (services) with other participants by contributing them in a controlled policy-driven manner to the collaboration. To make cloud computing truly scalable and to support interoperability issues, a DC platform among CPs is very important.

A dynamic collaborative cloud service platform can help CPs to maximize their profits by offering existing service capabilities to collaborative business partners. These capabilities can be made available and tradable through a service catalog for easy exchange to provide new value-added collaborative cloud services to consumers. Furthermore, the DC platform can enable a CP to handle cloud bursting by redirecting some of the load to its collaborators. Figure 1 shows a dynamic collaborative cloud service platform.

The formation of a DC is initiated by a CP, which recognizes a good business opportunity in forming a DC with other CPs in order to provide a set of services to consumers.

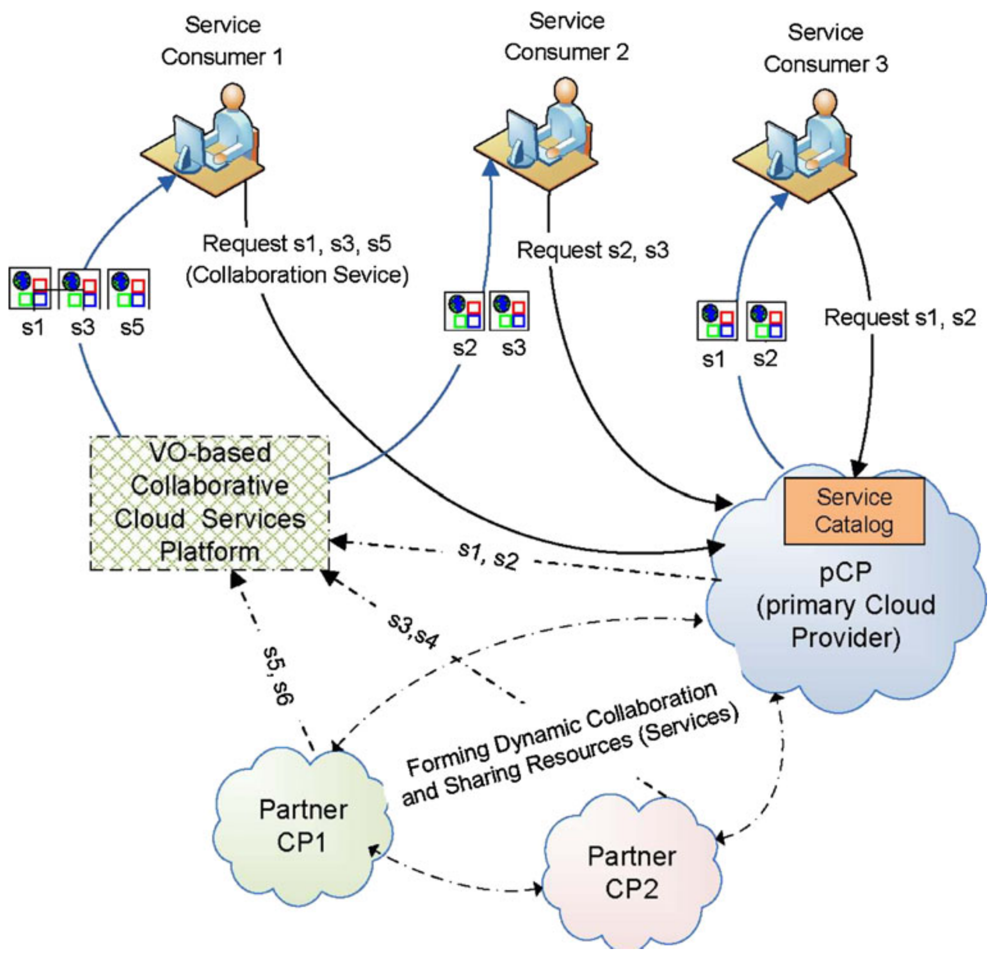


Fig. 1 A VO-based cloud services collaboration platform

The initiator is called the primary CP (pCP), while other CPs who share their resources/services in a DC are called collaborating or partner CPs. Users interact transparently with the VO-based DC platform by requesting services through a service catalog of the pCP. The CPs offer capabilities/services to consumers with a full consumption specification formalized as a standard SLA. The requested service requirements (single, multiple, or collaborative cloud services) are served either directly by the pCP or by any collaborating CP within the DC. Suppose that a pCP can provide two services s1 and s2 and CP1 and CP2 can provide services s3, s4, and s5, s6, respectively, as shown in Fig. 1. The request for collaborative services s1, s3, s5 or s2, s3 can be served by a VO-based DC platform. In case of services s1 and s2, the pCP can directly deliver the services. To enable and commercialize this DC platform, a CACM model is described in the next section.

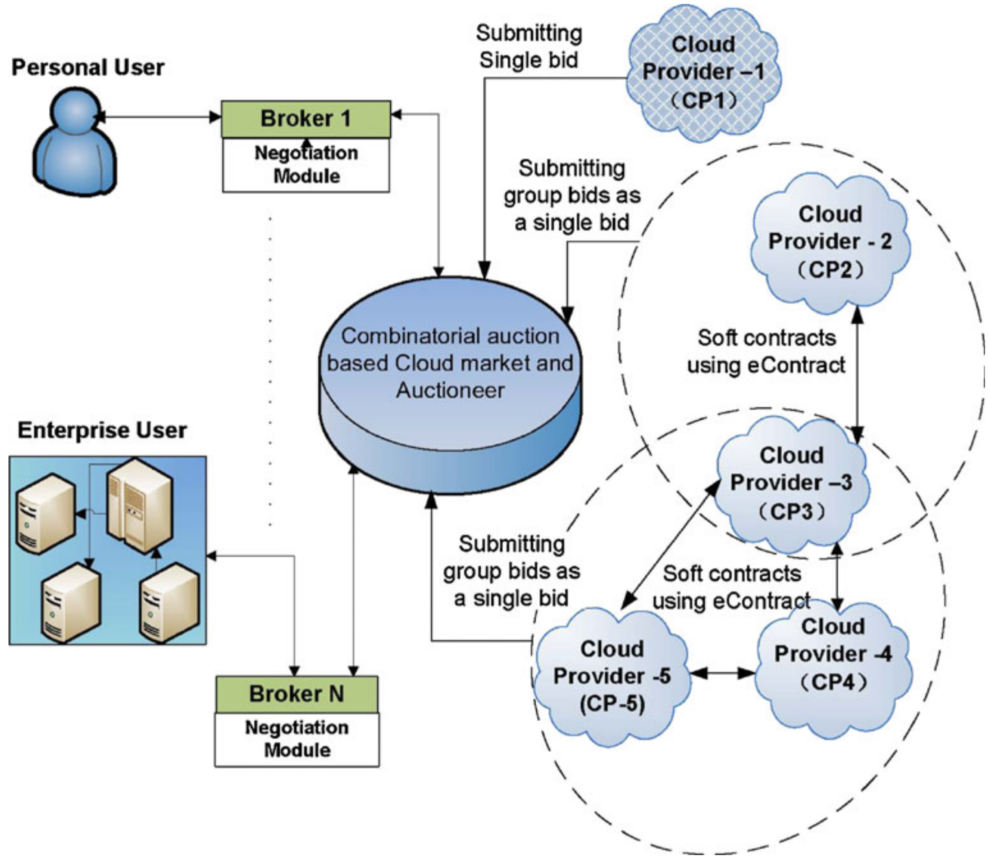
4 Proposed CACM model to facilitate a DC platform

4.1 Market architecture

The proposed CACM model to enable a DC platform among CPs is shown in Fig. 2. The existing auction policy of the CA is modified in the CACM model to address the issue of conflict minimization among providers in a DC platform. The existing and new auction policies for the CA model are shown in Figs. 3 and 4, respectively. The CACM model allows any CP to dynamically collaborate with appropriate partner CPs to form groups and to publish their group bids as a single bid to completely fulfill the consumer service requirements while also allowing the other CPs to submit bids separately for a partial set of services. We use the auction scheme based on [33] and [34] to address the CACM model. The main participants in the CACM model are brokers, users/consumers, cloud service providers, and auctioneers, as shown in Fig. 2.

Brokers in the CACM model mediate between consumers and CPs. A broker can accept requests for a set of services or composite service requirements from different users. A broker is equipped with a negotiation module that is informed by the current conditions of the resources/services and the current demand for its decisions. Consumers, brokers, and CPs are bound to their requirements and related compensations through SLAs. Brokers gain their utility by addressing the difference between the price paid by the consumers for the computing resources and that paid to the CPs for leasing their resources.

Fig. 2 Proposed CACM model to enable a DC platform among CPs



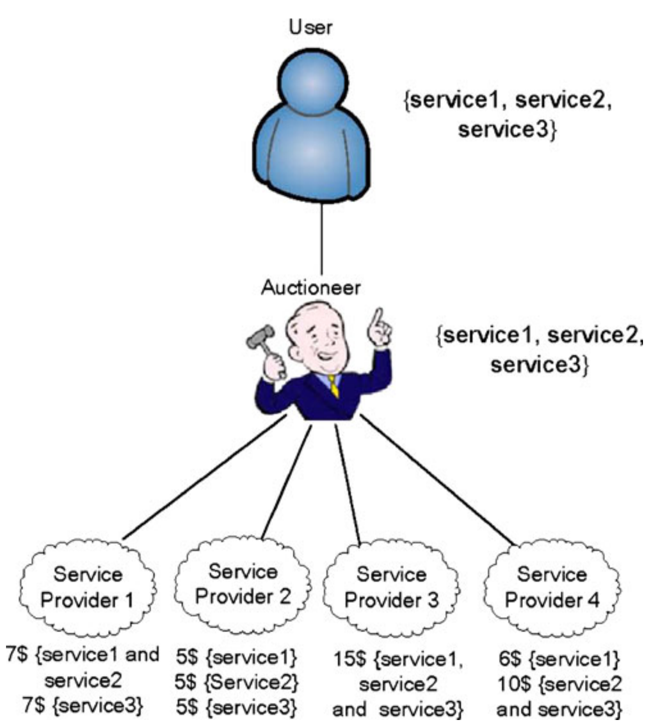


Fig. 3 Existing auction policy of CA

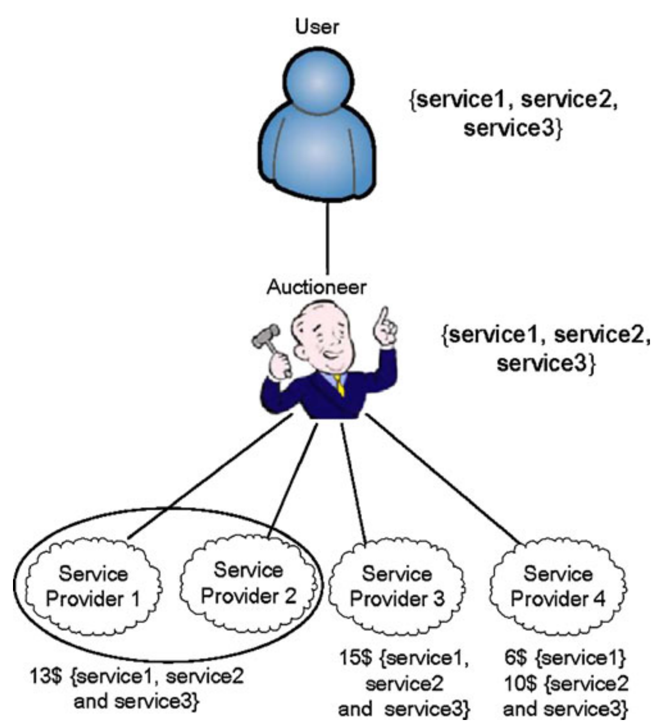


Fig. 4 New auction policy in CACM

The users/consumers can be enterprises or individual users. Consumers have their own utility functions that include factors such as deadlines, fidelity of results, and turnaround times of applications. They are also constrained by the amount of resources that they can request at any time, usually by a limited budget. The users can bid a single price for different composite/collaborative cloud services provided by CPs.

The CPs provide cloud services/resources including computational power, data storage, software-as-service (SaaS), computer networks, or infrastructure-as-a service (IaaS). A CP participates in an auction based on its interest and profit. It can publish a bid separately or collaboratively with other partner CPs by forming groups to fulfill the consumers' service requirements.

The responsibility of an auctioneer includes establishing the rules of the auction and conducting the combinatorial auction. The auctioneer first collects bids (single or group bids) from different CPs participating in the auction and then decides the best combination of CPs who can meet user requirements for a set of services using a winner determination algorithm. We utilize a secured generalized Vickrey auction [33] to address the CACM model problem and use dynamic graph programming [34] for the winner determination algorithm.

4.2 Additional components of a CP to form a DC platform in the CACM model

To achieve DC using the CACM model, in our architecture, a CP should possess these additional components (along with other components mentioned in [3]):

Price Setting Controller (PSC): A CP is equipped with a PSC which sets the current price for the resource/service based on the market conditions, user demand, and current level of utilization of the resource. Pricing can be either fixed or variable depending on the market conditions.

Admission and Bidding Controller (ABC)—The ABC selects the auctions in which to participate and submits a single or group bid based on an initial estimate of the utility. Market information from the information repository is required to make decisions about which auctions to join.

Information Repository (IR)—The IR stores the information about the current market condition, different auction results, and consumer demand. It also stores the INIs (price, quality of service, reliability, etc.) and PRIs (past collaboration experiences) of other CPs, as well as market and consumer feedback about their services.

Collaborator Selection Controller (CSC)—The CSC helps a CP to find a good combination of collaborators to completely fulfill the consumer requirements by running a MOGA called MOGA-IC (described later in Section 5.3) utilizing the INI and PRI of other CPs.

Mediator (MR)—The (resource) mediator within a DC guarantees that the participating CPs are able to conform to changing circumstances and are able to accomplish their objectives in a dynamic and uncertain environment. Once a DC platform is established, the mediator controls which resources/services are offered, how this decision is implemented, and which policies are being used. A mediator holds the initial policies for DC formation, creates an eContract, and negotiates with other CPs through its local collaborating agent (CA).

Service Registry (SR)—The SR encapsulates the resource and service information for each CP. In the case of a DC, the SR is accessed by the MR to acquire the necessary local resource/service information. When a DC is created, an instance of the SR is created that encapsulates all local and delegated external CP partners' resources/services.

Policy Repository (PR)—the PR virtualizes all of the policies within a DC platform, including the MR policies and the DC creation policies, along with any policies for resources/services delegated to the DC due to a collaborating arrangement. These policies form a set of rules to administer, manage, and control access to DC resources and also to help rearrange cloud services, providing a way to manage components in the face of complex technologies.

Collaborating Agent—the CA is a policy-driven resource discovery module for DC creation and is used as a conduit by the MR to exchange eContracts with other CPs. It is used by a primary CP to discover the collaborating CPs (external) resources/services, as well as to let them know about the local policies and service requirements prior to the actual negotiation by the MR.

4.3 Formation of a DC platform in the CACM model

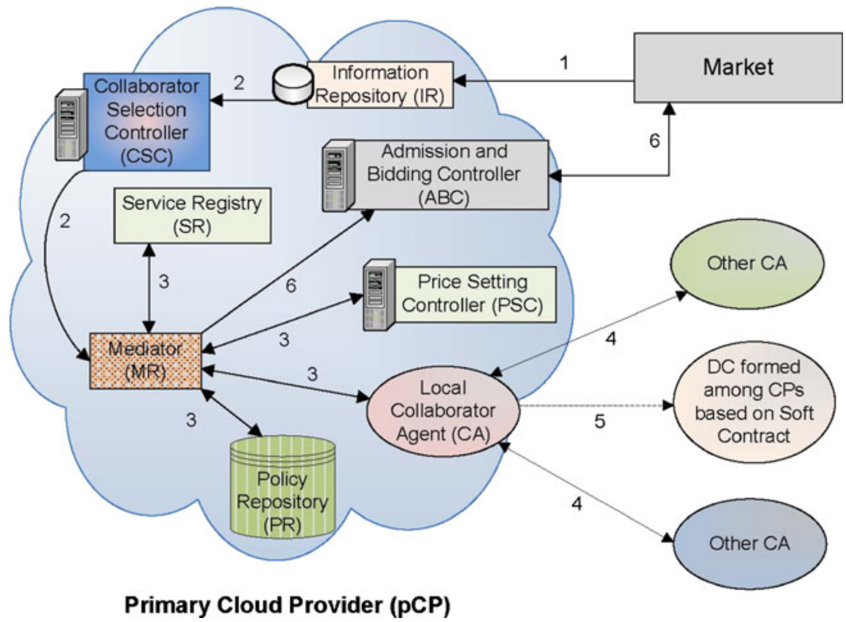
The DC creation steps are shown in Fig. 5 and are explained as follows:

Step 1 The IR is used to inform the pCP of a business opportunity in the market. The pCP cannot provide all of the service requirements of an auction set and so aims to form a collaborative group in order to address consumer requirements.

Step 2 The CSC is activated by the pCP to find a set of pareto-optimal solutions for partner selection, and it chooses any combination from the set to form groups and sends this information to the MR.

Step 3 The MR obtains the resource/service and accesses information from the SR on SLAs and other policies from the PR. It generates an eContract that encapsulates its service requirements based on the current circumstance, its own contribution

Fig. 5 The formation of a DC platform among CPs



policies, costs of services (generated by PSC), and SLA requirements of its customer(s), and passes this eContract to the local CA.

Step 4 The local CA of a pCP carries out negotiations with the CAs of other identified partner CPs using the eContract. Since the group members know each other very well, the number of conflicts will be reduced. When all CPs (including the pCP) agree with one other, they form a soft contract. A soft contract guarantees that resources/services will be available if the group wins the auction.

Step 5 When the pCP acquires all services/resources from its collaborators to meet the SLA of the consumer, a DC platform is formed. If no CP is interested in such arrangements, DC creation is resumed from Step 2 with another pareto-optimal solution group.

Step 6 Once the DC platform is established, the MR of the pCP submits collaborative bids as a single bid to the market using the ABC. If this group wins an auction bid, a hard contract is formed among group members to finalize the agreement in the DC. A hard contract ensures that the collaborating CPs will provide the resources/services according to the SLAs with consumers.

If some CPs win an auction separately for each service (few chances are available), they need to make a hard contract for DC creation, and, as we pointed out earlier (see Section 1), a large number of conflicts may occur among the participating CPs.

An existing DC may need to either demobilize or rearrange itself if any of the following conditions hold: (a) the circumstances under which the DC was formed no longer hold; (b) collaboration is no longer beneficial for the participating CPs; or (c) participating CPs are not meeting their agreed upon offerings.

4.4 System model for auction in the CACM

For convenience of analysis, the parameters and variables for the auction models are defined as follows:

$R=\{R_j|j=1\ldots n.\}$: a set of n service requirements of the consumer

$P=\{P_r|r=1\ldots m.\}$: a set of m CPs who participate in the auction as bidders

P_{rj}=a cloud provider r who can provide service j

$S(P_r)$=a set of services ($S_{j=1\ldots n}$) provided by any provider r where $S(P_r)\subseteq R$

$\Omega_{\max}(R, Q)$=a payoff function of the user where R is the service requirement and Q defines the SLAs of each service.

4.4.1 Single and group bidding functions of CPs

Let M be a service cost matrix of any provider P_r, S be any service in R (i.e., $S\subseteq R$), and G be a group of providers in P (i.e., $G\subseteq P$). To simplify the auction model, we assume that each CP can provide at most two services because it is not feasible for a CP to provide all kinds of services. The matrix M includes the costs of the provider's own services as well as the collaboration costs (CCs) between its own services and those of other providers. Figure 6 illustrates the matrix M. We assume that P_r provides two services, CPU and Memory. Let $a_{ii}(i=1,...,n)$ be the cost of providing any service in M independently, a_{ij} $(i, j=1,...,n,\ i\neq j)$ be the

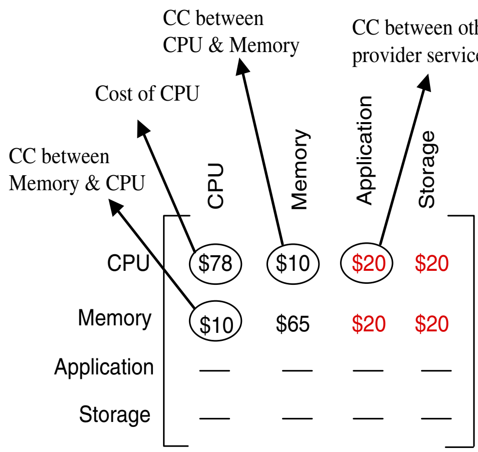


Fig. 6 Cost matrix M

CC between S_i and S_j services (S_i, $S_j \in S(P_r)$), and a_{ik}(i, k= 1,...,n, $i \neq k$) be the CC between S_i and S_k services ($S_i \in S(P_r)$ and $S_k \notin S(P_r)$). We set a nonreciprocal CC between $S(P_r)$ services in M which is practically reasonable.

If provider P_r has past collaboration experience, it can store true CC for services with other providers. Otherwise, it can set a high CC for other providers. The CCs of services with other providers in matrix M are updated when the providers finish a negotiation and collaboratively provide the services to consumers in the DC platform.

Assume now that the *Bidding Function* of any provider P_r submits a bid separately to partially fulfill the customer service requirements without collaborating with other CPs. This bidding function can be determined as follows: $\phi_{(SP_r)} = C_{S(P_r)} + \gamma(P_r)$ where $C_{S(P_r)}$ is the total cost incurred by the provider P_r to provide $S(P_r)$ services ($S(P_r) \subseteq R$), and $\gamma(P_r)$ is the expected profit of the provider P_r. The total cost $C_{S(P_r)}$ is calculated as follows using the matrix M:

$$C_{S(P_r)} = \sum_{S_i \in S(P_r)} a_{ii} + \sum_{S_i \in S(P_r)} \sum_{S_j \in S(P_r)} a_{ij} + \sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_r)} a_{ik} \tag{1}$$

where, i, j, k=1,...,n and $i \neq j \neq k$

The first term in Eq. 1 is the cost of providing services $S(P_r)$.The second term is the total collaboration cost between $S(P_r)$ services, and the third term refers to the total collaboration cost between services of different CPs with whom provider P_r collaborates. Since provider P_r does not know with whom it will collaborate after winning an auction, the true cost of a_{ik} cannot be determined. Therefore, P_r may set a high collaboration cost in a_{ik} in order to avoid potential risk in the collaboration phase.

Now the *Bidding Function* of a group of CPs who submit their bids collaboratively as a single bid to completely fulfill the service requirements can be determined as follows: Let P_r form a group G by selecting appropriate partners where $S(P_G)$ is the set of services provided by G and $S(P_G) \subseteq R$, $G \subseteq P$. For any provider such as $P_r \in G$, the total cost of providing $S(P_r)$ services is

$$C^G_{S(P_r)} = \sum_{S_i \in S(P_r)} a_{ii} + \sum_{S_i \in S(P_r)} \sum_{S_j \in S(P_r)} a_{ij} + \sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig} + \sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_G)} a_{ik} \tag{2}$$

where, i, j, g, k=1,...,n and $i \neq j \neq g \neq k$

We can see from Eq. 2 that the term $\sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_r)} a_{ik}$ of Eq. 1 is now divided into two terms in $C^G_{S(P_r)}$: $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$ and $\sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_G)} a_{ik}$. The term $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$ denotes the total collaboration cost of services of provider P_r with other providers in the group. The term $\sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_G)} a_{ik}$ refers to the total collaboration cost for services of other CPs outside of the group with whom provider P_r needs to collaborate. This term can be zero if the group can satisfy all of the service requirements of the consumer. Since P_r knows other group members, it can find the true value of the term $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$. Moreover, if P_r applies any good strategy to form the group G, it is possible for P_r to minimize $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$. Hence, this group G is more likely to win the auction than other providers who submit separate bids to partially fulfill the service requirements. So the *Bidding Function* for group G can be calculated as follows:

$$\phi^G_{S(P_G)} = \sum \left(C^G_{S(P_r)} + \gamma^G(P_r) \right), \forall P_r \in G, \quad r = 1, \ldots, l \tag{3}$$

where l is the number of providers in G and $\gamma^G(P_r)$ is the expected profit of any provider r in the group.

4.4.2 Payoff function of the user/consumer

With the help of a broker, a user generates the payoff function. During an auction, the user uses the payoff function $\Omega_{\max}(R, Q)$ to internally determine the maximum payable amount that it can spend for a set of services. If the bid price of any CP is greater than the maximum payable amount $\Omega_{\max}$, it will not be accepted. In the worst case, an auction terminates when the bids of all CPs are greater than $\Omega_{\max}$. In such a case, the user modifies its payoff function, and the auctioneer reinitiates the auction with a modified payoff function.

4.4.3 Motivations of the CPs to form a group

Let $\phi^G_{S(P_r)}$ be the price of the provider r when it forms a group G where $C^G_{S(P_r)}$ is the cost of its services in the group. The expected profit for the P_r in the group is $\gamma^G(P_r) = \phi^G_{S(P_r)} - C^G_{S(P_r)}$. We know that the expected profit for provider *r, who submits a bid separately, is* $\gamma(P_r) = \phi_{S(P_r)} - C_{S(P_r)}$. We argue that if any CP forms a group using a good partner selection strategy, it can increase its profit over what it would have received by separately publishing the bid. To calculate the increased profit, we consider the following assumptions:

$C^G_{S(P_r)} \leq C_{S(P_r)}$ and $\gamma^G(P_r) = \gamma(P_r)$

Since provider r can collaboratively publish the bid, it may minimize its collaboration cost by selecting good partners, that is, $C^G_{S(P_r)}$ should be less than or equal to $C_{S(P_r)}$. However, $\gamma^G(P_r)$=$\gamma(P_r)$ means that the expectation of profit does not change. Consequently, we can also deduce the following:

$$\phi^G_{S(P_r)} \leq \phi_{S(P_r)} \tag{4}$$

The provider who collaboratively publishes a bid can provide lower prices for its services while maintaining the same expected profit. Thus, it has more chances to win the auction. To determine the increased profit for P_r, let $\phi^{2\mathrm{LP}}_{S(P_r)}$ be the second lowest price that will be paid to P_r for $S(P_r)$ services if it wins the auction. Now if P_r attends any auction and applies separate and collaborative bidding strategies alternatively, the increased profit $\gamma^I(P_r)$ for P_r can be calculated as follows:

$$\gamma^I(P_r) = \alpha\left(\phi^{2LP}_{S(P_r)} - C^G_{S(P_r)}\right) - \beta\left(\phi^{2LP}_{S(P_r)} - C_{S(P_r)}\right) \tag{5}$$

where

α= { 1 if provider r collaboratively wins the auction}0 otherwise}.
β= { 1 if provider r separately wins the auction}0 otherwise}.

From Eq. 5, we observe that if provider r collaboratively wins an auction, it should always increase its profit. Otherwise, no increased profit will be achieved. A good partner selection strategy is required for a CP to form a group. In the next section, we describe an effective MO optimization model for a good method of partner selection.

5 Model of partner selection

5.1 Partner selection problem

A primary/initiator CP (pCP) identifies a business opportunity which is to be addressed by submitting a bid for a set of consumer services. It needs to dynamically collaborate with one or more CP partners in order to satisfy the consumer service requirements completely since it cannot provide all of the services on its own. We assume that each CP can provide one or at most two services, and each service has one or more providers. Furthermore, each CP can participate in other groups simultaneously. This process of CP partner selection is shown in Fig 7.

Figure 7 shows that the pCP ($P_{1,1}$) can provide s1 service and needs four other CP partners among 12 candidate CP partners to provide a total of five kinds of consumer service requirements (s1, s2, s3, s4, and s5). We also assume that the pCP has the INIs and PRIs of all of the other providers for each service. The INI includes price and quality information for the services of other providers, the most important factors. The PRI includes the number of projects/auctions accomplished/won by other providers among themselves and with the pCP. The pCP can obtain all of this information from each CPs website, from the market and also from consumers' feedback about their services.

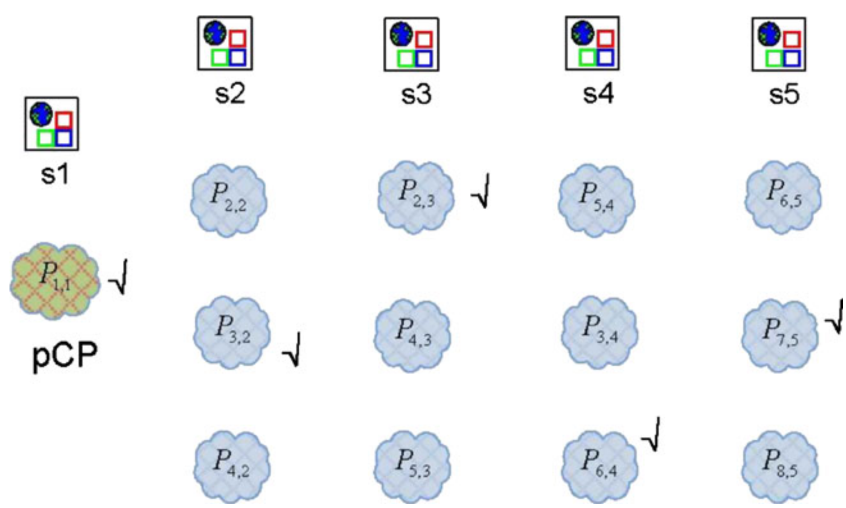


Fig. 7 Partner selection process for the pCP

5.2 MO optimization problem for partner selection

The parameters for MO partner selection are defined as follows:

ϕ_{rj}=price of CP r for providing service j independently
Q_{rj}=quality value for service j of CP r (qualitative information can be expressed by the assessment values from 1 to 10 (1 very bad, 10 very good))
$W_{rj,xi}$=the value of past collaboration experience (i.e., the number of times providers have collaboratively won an auction) between provider r for service j and another provider x for service i (r, x=1,...,m; i, j=1,...,n; $i \neq j$) and $U = \{U_{rj} | r = 1.....m, j = 1...n\}$: a decision vector for partner selection where $U_{rj} = \begin{cases} 1 \text{ if choose } P_{rj} \\ 0 \text{ otherwise} \end{cases}$ and

$U_{rj}U_{xi} = \begin{cases} 1 & \text{if choose } P_{rj} \text{ and } P_{xi} \\ 0 & \text{otherwise} \end{cases}$

The goal is to select a group of CP partners who repeatedly collaboratively win auctions (maximizing past relationship performance values) while minimizing cost and maximizing quality value. In most situations, no candidate provider group can meet all goals. To solve the partner selection problem of a pCP using the INIs and PRIs, an MO optimization model to minimize total price and maximize total collaborative past relationship (PR) performance and service quality values can be expressed mathematically as follows:

$$\text{Minimize Obj\_1} = \sum_{j=1}^{n} \sum_{r=1}^{m} \phi_{rj} U_{rj} \tag{6}$$

$$\text{Maximize Obj\_2} = \sum_{j=1}^{n} \sum_{r=1}^{m} Q_{rj} U_{rj} \tag{7}$$

$$\text{and Maximize Obj\_3} = \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \sum_{r,x=1}^{m} W_{rj,xi} U_{rj} U_{xi} \tag{8}$$

subject to

$$U_{rj} = \begin{cases} 1 & \text{if choose } P_{rj} \\ 0 & \text{otherwise} \end{cases}$$

$$U_{rj}U_{xi} = \begin{cases} 1 & \text{if choose } P_{rj} \text{ and } P_{xi} \\ 0 & \text{otherwise} \end{cases}$$

5.3 Multi-objective genetic algorithm

In the CP partner selection problem, MO optimization is preferable because it provides a decision-maker (pCP) with several trade-off solutions. Actually, the CP partner selection problem has multiple conflicting objectives, minimizing the price of service while maximizing past relationship performances and service quality values. Multiple objective formulations are practically required for concurrent optimization that yields optimal solutions to balance conflicting relationships among the objectives. MO optimization yields a set of pareto-optimal solutions, which is a set of solutions that are mutually non-dominated [28]. The concept of non-dominated solutions is required when comparing solutions in a multi-dimensional feasible design space formed by multiple objectives.

A solution is said to be pareto-optimal if it is not dominated by any other solution in the solution space. The set of all such feasible non-dominated solutions in a solution space is termed the pareto-optimal solution set. For a given pareto-optimal solution set, the curve made in the objective space is called the pareto front. When two conflicting objectives are present, there will always be trade-offs when moving from one pareto solution to another. A pareto-optimal solution set is often preferred over a single solution, because the set helps to elucidate the trade-offs among conflicting objectives and to make informed selections about the optimal solution.

MO optimization difficulties can be alleviated by avoiding multiple simulation runs, avoiding artificial aids such as weighted sum approaches, using efficient population-based evolutionary algorithms, and the concept of dominance. The use of MOGAs provides a decision-maker with the practical means to handle MO optimization problems. When solving PSP for CPs using MOGA techniques, one important issue needs to be addressed: the identification of an appropriate diversity preservation mechanism for selection operators that enhances the yield of pareto-optimal solutions during optimization, particularly for CP partner selection problems with multiple conflicting objectives. We developed MOGA-IC with two popular MOGAs, the NSGA-II [28] and the SPEA2 [29], both of which include an excellent mechanism for preserving population diversity in the selection operators. In this section, the MOGA-IC is designed for the proposed model of CP partner selection.

Natural number encoding is adopted to represent the chromosome of an individual. A chromosome of an individual is an ordered list of CPs. Let y=[y_1, y_2, …, y_j,…,y_n](j=1, 2,…,n), y_j be a gene of the chromosome, with its value between 1 and m (for service j, there are m CPs for a response). If m=50 and n=5, there may be ten CPs that can provide each service j. Thus, a total of 10^5 possible solutions are available. In this way, the initial populations are generated. A two-point crossover is employed, and in the case of mutation, one provider is randomly changed for any service. For the selection operator in the proposed MOGA-IC, NSGA-II, and SPEA2 are implemented. The complete procedures for developing the MOGA-IC with NSGA-II and SPEA2 are given below.

Algorithm 1 NGSA-II

Step 1 Initialize the input parameters which contain the number of requirements (R), providers (m) and maximum genetic generations (G), population size (N), crossover probability (p_c), and mutation probability (p_m).

Step 2 Generate the initial parent population P_t (t=0) of size N_P.

Step 3 Apply a binary tournament selection strategy to the current population and generate the offspring population O_t of size N_O=N_P with the predetermined p_c and p_m.

Step 4 Set S_t=P_t∪O_t, apply a non-dominated sorting algorithm, and identify different fronts F_1, F_2,..., F_a.

Step 5 If the stop criterion (t>G) is satisfied, stop and identify the individuals (solutions) in population P_t and their corresponding objective values as the pareto (approximate)-optimal solutions and pareto-optimal fronts.

Step 6 Set the new population P_{t+1}=0. Set the counter i=1 until |P_{t+1}|+|F_i|≤N, set P_{t+1}=P_{t+1}∪F_i and i=i+1.

Step 7 Perform the crowding-sort procedure and include the most widely spread (N−|P_{t+1}|) solutions found using the crowding distance values of the sorted F in P_{t+1}.

Step 8 Apply binary tournament selection, crossover, and mutation operators to P_{t+1} to create the offspring population O_{t+1}.

Step 9 Set t=t+1, then return to Step 4.

Algorithm 2 SPEA2

Step 1 Generate a random population P_0 of size N_P. Set t=0 and generate an empty external archive E_0 of size N_E.

Step 2 Calculate the fitness of each solution x in P_t∪E_t as follows:

Step 2.1 Calculate the raw fitness as $R(x,t)=\sum_{y\in P_t\cup E_t, y\succ x} S(y,t)$ where $S(y, t)$ is the number of solutions in P_t∪E_t dominated by solution y.

Step 2.2 Calculate the density as $D(x,t)=\left(\sigma_x^k+2\right)^{-1}$, where σ_x^k is the distance between solution x and its kth nearest neighbor, where $k=\sqrt{N_p+N_E}$.

Step 2.3 Assign a fitness value as $F(x, t)$=$R(x, t)$+$D(x, t)$.

Step 3 Copy all non-dominated solutions in P_t∪E_t to E_{t+1}.

Now, two cases may arise. Case 1: if |E_{t+1}|>N_E, then truncate |E_{t+1}|−N_E solutions by iteratively removing solutions that have maximum σ^k distances. Break any tie by examining σ^l for l=k−1,..,1 sequentially. Case 2: if |E_{t+1}|≤ N_E, copy the best N_E−|E_{t+1}| dominated solutions according to their fitness values from P_t∪E_t to E_{t+1}.

Step 4 If the stopping criterion is satisfied, stop and copy the non-dominated solutions in E_{t+1}.

Step 5 Select the parent from E_{t+1}using binary tournament selection with replacement.

Step 6 Apply the crossover and mutation operators to the parents to create N offspring solutions. Copy offspring to P_{t+1}, t=t+1, then return to Step 2.

6 Evaluation

In this section, we present our evaluation methodology and simulation results for the proposed CACM model with the new auction policy and the MOGA-IC for CP partner selection. First, we compare the proposed CACM model with the existing CA model in terms of economic efficiency. Then we present a simulation example of PSP for a pCP in the CACM model, used to illustrate the proposed MOGA-IC method. NSGA-II and SPEA2 are utilized to develop the MOGA-IC. Further simulation examples are conducted to pinpoint the most viable approach (NSGA-II or SPEA2) for MOGA-IC. Moreover, we implement the existing MOGA that uses only INI, MOGA-I, for CP partner selection and analyze its performance with MOGA-IC in the proposed CACM model. We implement the CACM model (winner determination algorithm) with the new auction policy as well as the MOGA-IC in Visual C++.

6.1 Evaluation methodology

One of the main challenges in the CACM model and the PSP of CP is the lack of real world input data. We conduct the experiments using synthetic data. We generate the input data as follows:

Many CPs (m = 100) with different services and some consumer requirements (R = 3–10) are generated randomly. We assume that each CP can provide at most two services so that they have to collaborate with others to fulfill the service requirements R. Each service may have one or more CP. Based on R, CPs are selected. It is possible that every CP may not provide the required R. Also the cost of providing any independent service is randomly generated from \$80 to \$100. The ranges of CCs of services as well as the profit are set within \$10–\$30 and \$10–\$20, respectively. Quality and collaborative performance values of the providers are randomly selected from 1 to 10 and 0 to 10, respectively. If any provider has more collaboration experience with other providers, the CC can be minimized. We use the following formula to calculate the CC between any providers P_{rj} and P_{xi}:

$$\mathrm{CC}_{rj,xi} = \mathrm{CC}_{\min} + (CC_{\max} - \mathrm{CC}_{\min}) \times \frac{1}{e^{W_{rj,xi}}} \quad (9)$$

where

$\mathrm{CC}_{\min}$=the minimum CC between services (here \$10)
$\mathrm{CC}_{\max}$=the maximum CC between services (here \$30)
$W_{rj,xi}$=the number of collaboration experiences between P_{rj} and P_{xi}. If it is zero, the highest CC is set between the providers. Thus, the final price of services is generated for each provider, and it is varied based on the CC in different auctions.

6.2 Simulation results

6.2.1 Economic efficiency of the CACM model as compared to that of the existing CA model

As we have mentioned earlier (in Sections 1 and 2), the existing auction policy of the CA-based market model is not suitable to meet the requirements of a DC platform. One reason is that it cannot address the issue of conflict minimization among the CPs. Another reason is that when using this policy, the providers may include high CCs with their bidding prices as they do not know with whom they will need to collaborate after winning an auction. Thus, the total price for consumers, as well as the negotiation time, increases (up to the maximum price he/she can pay).

In contrast, the proposed auction policy in the CACM model allows any CP to dynamically collaborate with appropriate partner CPs to form groups before joining the auction and to publish their group bids as a single bid in order to completely fulfill the consumer service requirements. This new approach enables CPs to minimize conflicts and to calculate the true CCs with respect to one another.

This new approach is beneficial to the consumer or customer as the total price of the services decreases. To show the economic efficiency of the CACM model compared to that of the existing CA model, in our simulation, 1,000 auctions are generated for different consumer requirements. Based on those requirements, providers separately publish their bids to the existing CA market and also collaboratively publish their bids to the CACM market. The winners and final prices are determined by the auctioneers in both markets. After every 200 auctions, we count the prices of the winning bids that were determined by the existing CA market and the proposed CACM, respectively. Figure 8 shows the economic efficiencies of the two auction-based markets.

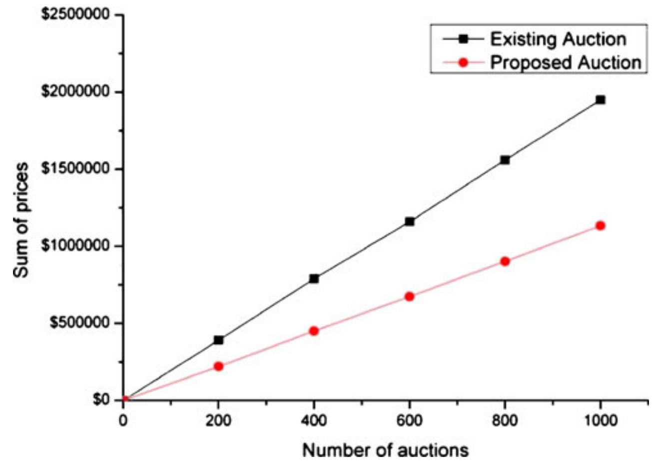


Fig. 8 Economic efficiency of the CACM model as compared to that of the existing CA model

It can be seen from Fig. 8 that when the number of auctions increases, the CACM auction model reduces the total service price to consumers as compared to the existing CA model for the same number of service requirements. The main reason is that CCs among the group members are lower and the total service price is reduced. A good partner selection algorithm like our proposed MOGA-IC is required to reduce the CCs as well as the conflicts among partner CPs.

6.2.2 Appropriate approach to develop the MOGA-IC

In this section, we present three simulation examples of PSP for a pCP in the CACM model. Table 1 shows the three simulation examples with MOGA-IC parameters for PSP in the CACM model. For each simulation example, MOGA-IC is developed based on NSGA-II and SPEA2. Furthermore, in each simulation example, two INIs (price and quality of services) and one PRI (number of auctions collaboratively won by other providers among themselves and also with pCP) of candidate CPs are considered. Both pieces of information are presented in Tables 2 and 3 in normalized forms for the first simulation example. For normalization, the method proposed by Hwang and Yoon [35] is utilized. We can see that 21 total CPs are found from 35 candidate CPs who can provide five randomly generated

Table 1 Three simulation examples with MOGA-IC parameters

Simulation examples	m	R	N/E	G	P_c	P_m
1	35	5	50	20	0.9	0.1
2	100	5	100	50	0.9	0.1
3	100	5	100	100	0.9	0.1

Table 2 The normalized INIs of the pCP and the other candidate CPs

Service no.	Provider no.	Price of service	Quality value of service
2	28	0.17	0.99
2	10	038	0.88
3	10	0.99	0.88
3	15	0.81	0.3
3	6	0.66	0.01
3	32	0.07	0.65
3	14	0.55	0.23
3	20	0.88	0.72
4	9	0.00	0.54
4	33	0.17	0.4
4	18	0.84	0.62
4	17	0.89	0.02
4	34	0.5	0.66
4	26	0.57	0.00
7	**1**	**0.4**	**1**
8	2	0.83	0.19
8	21	0.73	0.48
8	11	0.63	0.06
8	23	0.94	0.22
8	19	0.81	0.63
8	32	0.88	0.82

Table 3 The normalized PRIs of the pCP and the other candidate CPs

	P 28, 2	P 10, 2	P 10, 3	P 15, 3	P 6, 3	P 32, 3	P 14, 3	P 20, 3	P 9, 4	P 33, 4	P 18, 4	P 17, 4	P 34, 4	P 26, 4	*P 1, 7*	P 2, 8	P 21, 8	P 11, 8	P 23, 8	P 19, 8	P 32, 8
P 28, 2	–	–	0.51	0.5	0.79	0.25	0.79	0.14	0.69	0.66	0.24	0.54	0.18	0.3	0.29	0.28	0.36	0.42	0.96	0.97	0.72
P 10, 2	–	–	0.28	0.62	0.7	0.52	0.51	0.48	0.31	0.0	0.81	0.22	0.74	0.94	0.79	0.17	0.4	0.03	0.4	0.39	0.77
P 10, 3	–	–	–	–	–	–	–	–	0.49	0.67	0.28	0.13	0.41	0.63	0.93	0.66	0.17	0.01	0.70	0.26	0.96
P 15, 3	–	–	–	–	–	–	–	–	0.65	0.57	0.04	0.98	0.18	0.08	0.13	0.83	0.66	0.84	0.63	0.20	0.23
P 6, 3	–	–	–	–	–	–	–	–	0.48	0.38	0.28	0.18	0.38	0.27	0.81	0.11	0.77	0.79	1.0	0.29	0.96
P 32, 3	–	–	–	–	–	–	–	–	0.39	0.04	0.09	0.84	0.87	0.35	0.79	0.16	0.43	0.87	0.11	0.80	0.25
P 14, 3	–	–	–	–	–	–	–	–	0.68	0.54	0.29	0.32	0.21	0.44	0.85	0.09	0.18	0.666	0.19	0.52	0.74
P 20, 3	–	–	–	–	–	–	–	–	0.05	0.41	0.81	0.33	0.04	0.01	0.90	0.28	0.0	0.03	0.67	0.82	0.01
P 9, 4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.17	0.51	0.56	0.26	0.07	0.56	0.93
P 33, 4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.07	0.22	0.50	0.59	0.87	0.16	0.77
P 18, 4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.68	0.57	0.41	0.91	0.88	0.04	0.87
P 17, 4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.64	0.21	0.50	0.04	0.73	0.02	0.67
P 34, 4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.51	0.09	0.11	0.13	0.75	0.49	0.77
P 26, 4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.27	0.32	0.68	0.57	0.31	0.07	0.05
P 1, 7	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.22	1.0	0.8	0.47	0.58	0.96

Table 4 Pareto-optimal solutions of MOGA-IC with NSGA-II for example 1

Pareto-optimal solutions					Optimal objective function values		
$y = (y_7\ y_4\ y_3\ y_2\ y_8)$					Obj_1	Obj_2	Obj_3
1	18	6	10	32	3.16	3.32	7.63
1	18	10	10	32	3.49	4.2	7.33
1	34	10	28	32	2.94	4.35	6.24
1	9	32	28	21	1.37	3.66	4.93
1	9	10	28	32	2.44	4.23	6.65
1	9	32	28	19	1.45	3.81	5.49
1	9	14	28	32	2.00	3.58	6.82
1	34	32	10	32	2.23	4.01	6.97
1	18	10	28	32	3.28	4.31	6.44
1	9	32	10	32	1.73	3.89	5.88
1	34	32	28	32	2.02	4.12	5.59
1	34	6	10	32	2.82	3.36	7.39
1	34	32	10	19	2.16	3.82	6.48
1	34	10	10	32	3.15	4.24	7.12
1	9	32	28	32	1.52	4.00	5.44
1	18	14	10	32	3.05	3.55	7.27

Table 5 Pareto-optimal solutions of MOGA-IC with SPEA2 for example 1

Pareto-optimal solutions					Optimal objective function values		
$y = (y_7\ y_4\ y_3\ y_2\ y_8)$					Obj_1	Obj_2	Obj_3
1	9	14	28	19	1.93	3.39	6.10
1	9	32	28	32	1.52	4.00	5.44
1	9	32	28	19	1.45	3.81	5.49
1	9	32	28	11	1.27	3.24	4.93
1	9	14	28	32	2.00	3.58	6.82
1	9	10	28	32	2.44	4.23	6.65
1	34	32	28	19	1.95	3.93	5.73
1	34	32	28	32	2.02	4.12	5.59
1	9	32	10	32	1.73	3.89	5.88
1	34	10	10	32	3.15	4.24	7.12
1	34	32	10	19	2.16	3.82	6.48
1	34	32	10	21	2.08	3.67	6.16
1	9	32	28	21	1.37	3.66	4.93
1	18	10	10	32	3.49	4.2	7.33
1	34	10	28	32	2.94	4.35	6.24
1	34	6	10	32	2.82	3.36	7.39

consumer service requirements. We assume that provider number 1 is the pCP who can provide service number 7. The number of generations G in the first simulation example is set to 20 since the example search space is quite small.

In solving the first simulation example problem of CP partner selection, the best pareto front among the ten trials of 20 generations is selected as the final solution. The 16 pareto-optimal solutions of the first front of MOGA-IC with NSGA-II and SEPA2 for simulation example 1 are presented in Tables 4 and 5, respectively. Graphical representations are shown in Fig. 9.

From Fig. 9, it is very difficult to compare the performances of MOGA-IC with NSGA-II and MOGA-IC with SPEA2 since the solution space is quite small. We conduct

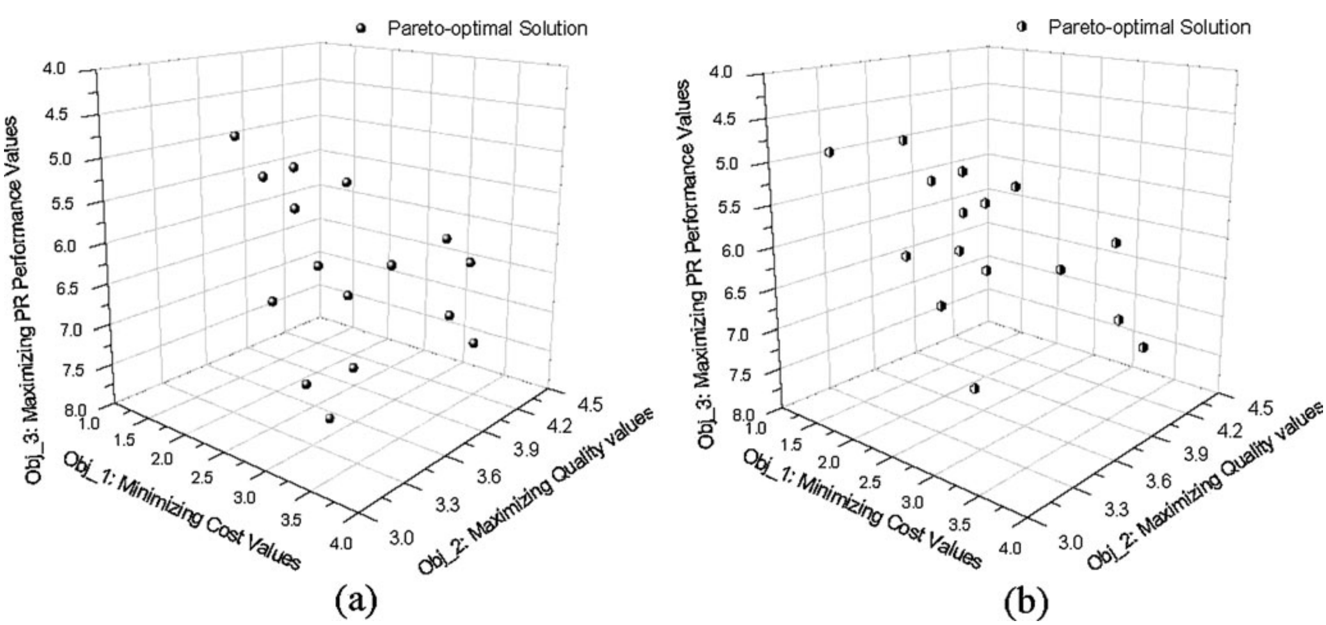


Fig. 9 Pareto-optimal solutions of MOGA-IC for simulation example 1 (N/E=50 and G=20) obtained by **a** NSGA-II and **b** SPEA2

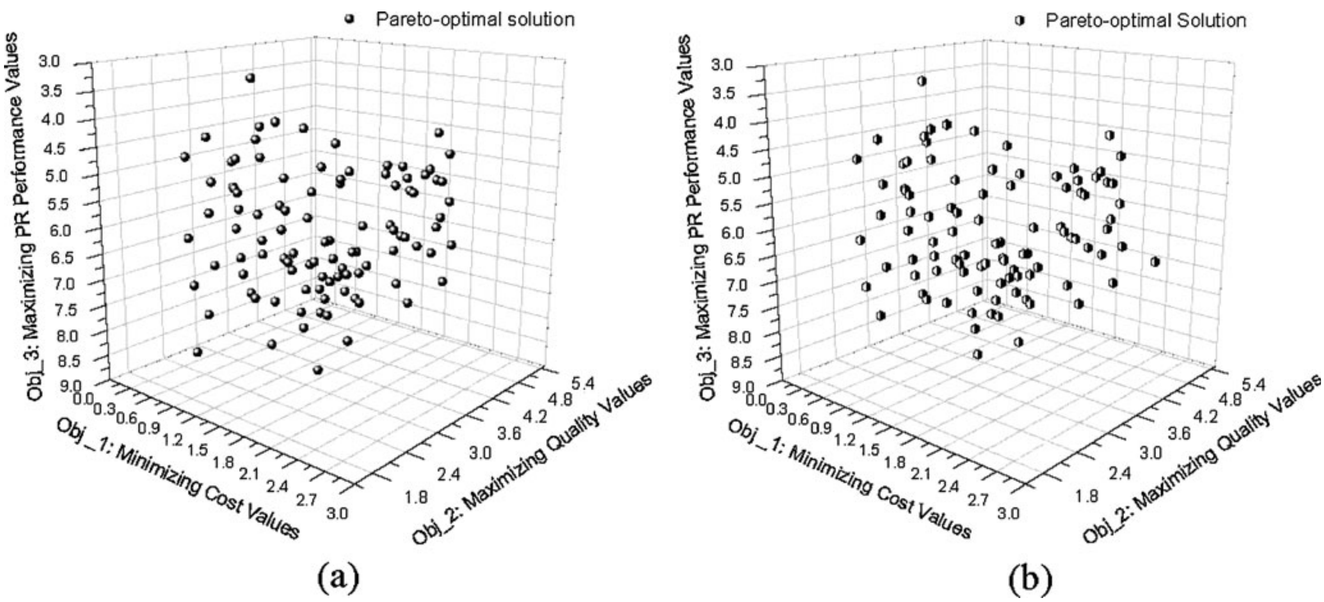


Fig. 10 Pareto-optimal solutions of MOGA-IC for simulation example 2 (N/E=100 and G=50) obtained by **a** NSGA-II and **b** SPEA2

further performance tests of MOGA-IC with NSGA-II and SPEA2 using simulation examples 2 and 3. Figures 10 and 11 show plots of pareto-optimal solution sets of the first fronts obtained by MOGA-IC using NSGA-II and SPEA2 when solving the simulation examples 2 and 3, respectively. Here, we only provide graphical representations of the pareto-optimal solutions for both the algorithms as the input data tables are very large.

In Figs. 10 and 11, the pareto fronts obtained using MOGA-IC with SPEA2 are dominated by MOGA-IC with NSGA-II solutions. To verify the inferior performance of SPEA2, Figs. 12 and 13 show the average optimized values of three objective functions in the first fronts during 50 and 100 generations using MOGA-IC with NSGA-II and MOGA-IC with SPEA2 for simulation examples 2 and 3, respectively.

It can be seen from Figs. 12 and 13 that SPEA2 initially finds better solutions quickly as compared to NSGA-II but in the end cannot provide the best solutions. Furthermore, the search direction in both algorithms is clearly visible in Figs. 12 and 13. For example, with SPEA2, the search direction is from high-cost to low-cost regions (Figs. 12a and 13a), while maintaining several

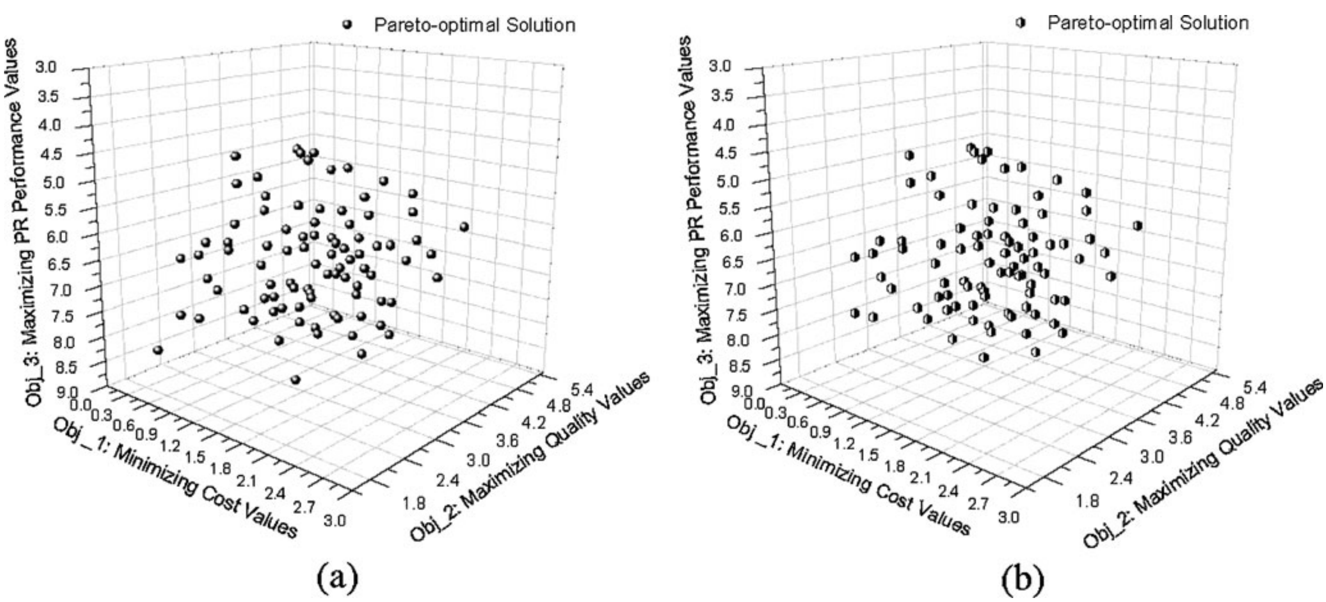


Fig. 11 Pareto-optimal solutions of MOGA-IC for simulation example 3 (N/E=100 and G=100) obtained by **a** NSGA-II and **b** SPEA2

extreme solutions on each generation's pareto front. In contrast, the NSGA-II pareto front moves toward the low-cost region without preserving each generation's extreme solutions. Instead, the entire pareto front shifts as new solution sets are obtained. In other words, MOGA-IC with SPEA2 yields pareto fronts with wider spans, while MOGA-IC with NSGA-II distributes solutions in a more focused manner due to the different selection strategies used by NSGA-II and SPEA2.

In MOGA-IC with NSGA-II, dominance ranking is used when forming the fronts of individuals, and these fronts are first used to populate the external set based on ranking, a strategy that allows a set of close-neighbor individuals in the same front to be included in the next generation. In contrast, MOGA-IC with SPEA2 selects individuals according to assigned fitness values based on Euclidean density information, so close-neighbor individuals are likely to be excluded in the next generation. The MOGA-IC with SPEA2 therefore yields pareto fronts with wider distributions of non-dominated solutions in contrast to MOGA-IC with NSGA-II, which is more focused when exploring the search space and generating pareto solution sets.

Next, consider the simulation runtimes of both MOGA-IC with NSGA-II and MOGA-IC with SPEA2, as shown in

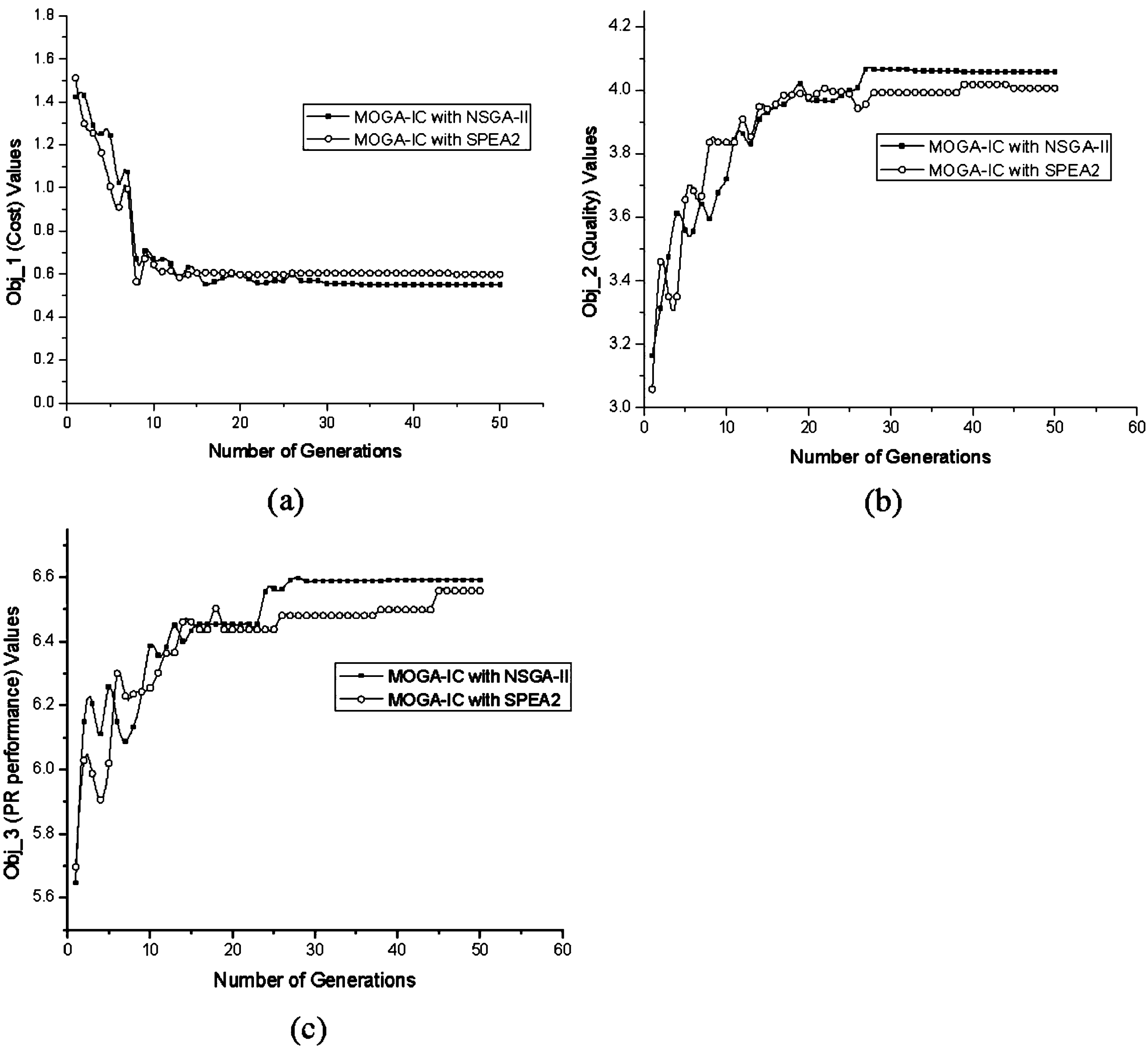


Fig. 12 Average optimized values of different objective functions in the first front of MOGA-IC with NSGA-II and SPEA2 for 50 generations. **a** Optimized average values of Obj_1 (cost) in NSGA-II and SPEA2. **b** Optimized average values of Obj_2 (quality) in NSGA-II and SPEA2. **c** Optimized average values of Obj_3 (PR performance) in NSGA-II and SPEA2

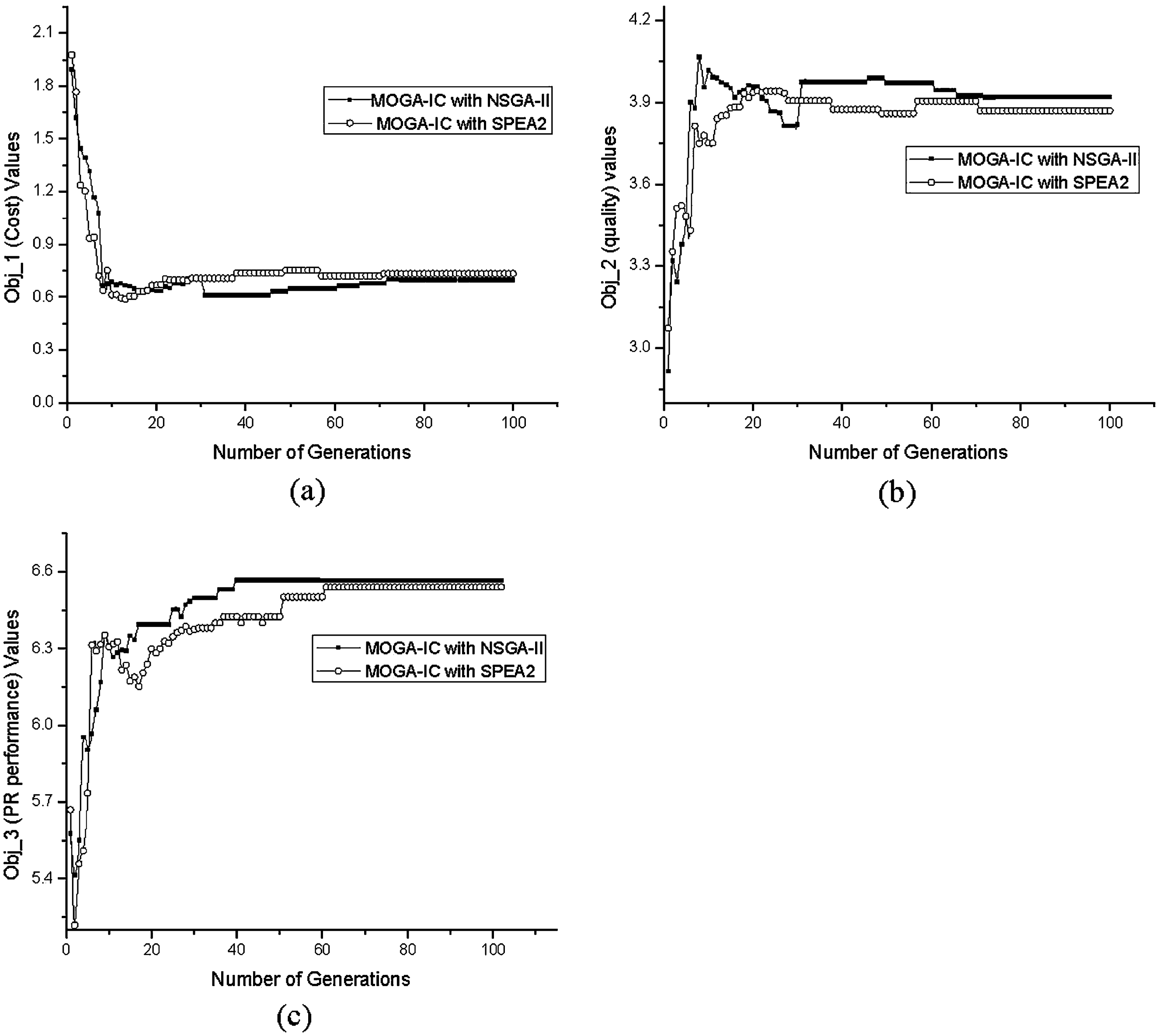


Fig. 13 Average optimized values of different objective functions in the first front of MOGA-IC with NSGA-II and SPEA2 for 100 generations. **a** Average optimized values of Obj_1 (cost) in NSGA-II and SPEA2. **b** Average optimized values of Obj_2 (quality) in NSGA-II and SPEA2. **c** Average optimized values of Obj_3 (PR performance) in NSGA-II and SPEA2

Table 6. From Table 6, we can see that MOGA-IC with NSGA-II runs much faster than does MOGA-IC with SPEA2 for the three simulation examples. The reason for this behavior is the time consumption in the truncation approach. The time consumption of MOGA-IC with the NSGA-II truncation approach is much lower than that of MOGA-IC with SPEA2. This is mainly due to the superiority of the truncation approach in MOGA-IC with

Table 6 Simulation runtimes of the three examples with MOGA-IC parameters

Simulation examples	m	R	N/E	G	P_c	P_m	Runtime (ms)	
							NSGA-II	SPEA2
1	35	5	50	20	0.9	0.1	16.823	25.036
2	100	5	100	50	0.9	0.1	123.603	261.809
3	100	5	100	100	0.9	0.1	298.831	450.019

NSGA-II. The MOGA-IC with NSGA-II uses crowding distance as a truncation approach when the sizes of non-dominated solutions exceed the archive size. A crowding distance is the average distance of its neighbors along each of the objectives. The smaller is a solution's crowding distance, the more crowded is the area in which the solution may be located. NSGA-II only needs to sort all solutions on each objective, and so the time consumption of its truncation approach is not very sensitive to the number of non-dominated solutions. However, MOGA-IC with SPEA2 uses a truncation operator based on a nearest neighbor strategy, and the number of non-dominated solutions directly relates to the efficiency of the truncation approach in SPEA2. So we found that NSGA-II is the appropriate algorithm to develop MOGA-IC for the CP partner selection problem. Thus, the pCP can select any combination of CP partners from the pareto-optimal solution sets obtained from MOGA-IC based on NSGA-II.

6.2.3 Performance comparison of MOGA-IC with MOGA-I in the CACM model

In order to validate the proposed MOGA-IC model for CP partner selection in the CACM model, we develop another MOGA called MOGA-I based on NSGA-II that uses INIs for CP partner selection. We analyze the performances of the pCP that use both MOGA-IC and MOGA-I algorithms to make groups and to join various auctions in the CACM model. We assume that initially no collaborative information for other CPs is available to the pCP.

At the beginning of each auction, all providers including the pCP form several groups using MOGA-I and submit several group bids as single bids for a set of services to the auctioneer. The winner determination algorithm proposed in [34] is used to find the winners. Next, in the same auction with the same set of services, the winner determination algorithm is executed again, but this time ,the pCP uses the proposed MOGA-IC (others use a MOGA-I approach) to join the auctions and determine the winners. In our simulation, 1,000 auctions are generated for different user requirements. After each 100 auctions, we count the number of auctions won by the pCP using both algorithms. The experimental results are shown in Fig. 14.

It can be seen from Fig. 14 that using the MOGA-IC approach, pCP wins more auctions than it does using the MOGA-I approach. The reason is that the past collaborative performance values increase as the number of auctions increases, and as a result, the MOGA-IC finds a good combination of partners for pCP.

We also validate the performance of MOGA-IC to compare to that of MOGA-I in terms of conflict minimization among the CP providers. We assume that conflicts

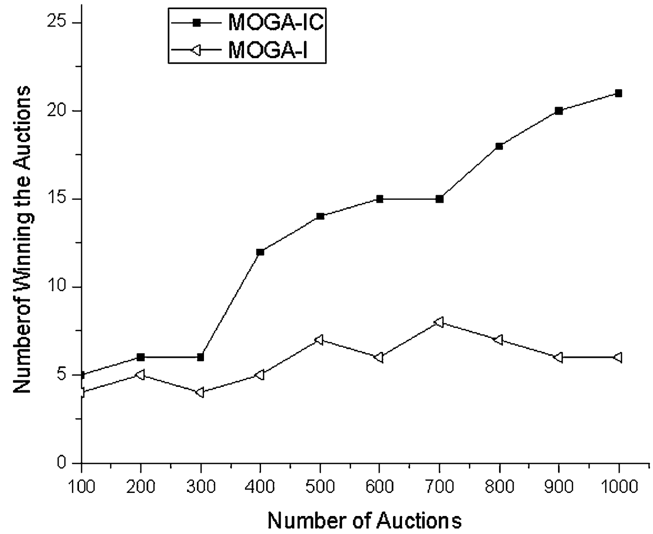


Fig. 14 Comparison of MOGA-IC with MOGA-I in terms of winning auctions

may happen between providers P_{rj} and $P_{xi}$with the probability

$$p_{\text{conflicts}} = \begin{cases} \frac{1}{\delta \times e^{W_{rj,xi}}}, & \text{if } W_{rj,xi} \neq 0 \\ \frac{1}{\delta} & \text{otherwise,} \quad \text{where } \delta \text{ is a constant} \end{cases} \quad i \neq j, r \neq x, \delta 1 \tag{10}$$

We set δ=20 assuming that there is a 5% chance of conflicts between any two providers P_{rj} and P_{xi} if they have no past collaborative experience. Like the previous experiment, 1,000 auctions are generated. For each auction, when pCP uses both algorithms and forms groups, we count the total number of conflicts that may happen among the group members for various services using the probability $p_{\text{conflicts}}$. The experimental results are shown in Fig. 15. We can see

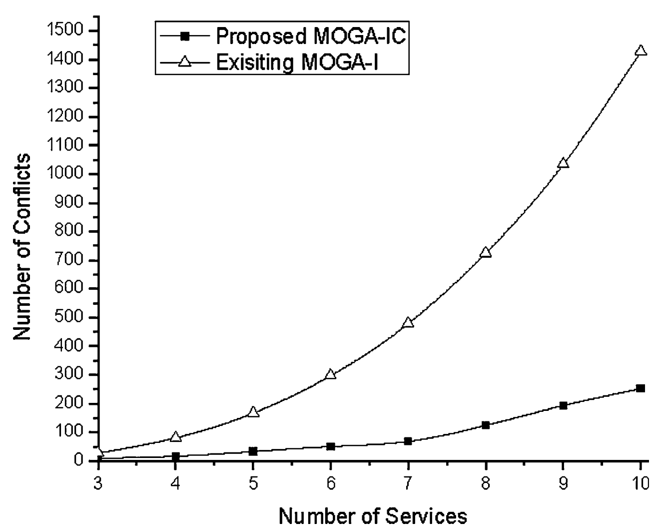


Fig. 15 Comparison of MOGA-IC with MOGA-I in terms of conflict minimization

from Fig. 15 that MOGA-IC can reduce a significant number of conflicts among providers as compared to the MOGA-I algorithm since it can utilize the PRI to choose partners along with the INI.

6.2.4 Scalability study of the MOGA-IC in terms of service requirements

In the proposed CACM model, for each consumer requirement (R), there is a separate combinatorial auction. Based on the consumer requirements, a pCP runs MOGA-IC to find appropriate CP partners to form groups. The MOGA-IC can support any number of consumer service requirements. To verify this, we conducted three simulations with MOGA-IC parameters as shown in Table 7.

From Table 7, we can see that the MOGA-IC average runtime is mainly related to three parameters, population size (N/E), maximum genetic generation (G), and consumer service requirements (R). When they increase, the runtime will become longer. However, the runtime increases minimally with increasing R when the other parameters are fixed, such as in examples 2 and 3 in Table 7.

7 Conclusions

This paper presents a novel combinatorial auction-based cloud market model called CACM to enable a DC platform among CPs which can fairly address interoperability and scalability issues for cloud computing. The CACM model uses a new auction policy that allows CPs to dynamically collaborate with other partners and to form groups and submit their group bids for a set of services as single bids. This policy can help to reduce collaboration costs as well as conflicts and negotiation time among CPs in DC and therefore creates more opportunities for the group to win auctions. A new multi-objective optimization model of partner selection using individual and past collaborative information is also proposed. An effective MOGA, MOGA-IC with NSGA-II, is developed to solve the model. The simulation results show that the MOGA-IC with NSGA-II is superior to MOGA-IC with SPEA2 for solving the partner selection problem of CPs. Compared with the existing MOGA-I approach, MOGA-IC with NSGA-II shows better performance results in CP partner selection as well as conflict minimization among CPs in the CACM model. In the future, we will try to simulate the proposed CACM model and the MOGA-IC with real world data to verify its economic efficiency and performance.

Table 7 MOGA-IC simulation results with various consumer service requirements

Simulation examples	m	R	N/E	G	P_c	P_m	Runtime (ms) MOGA-IC with NSGA-II
1	100	**6**	80	100	0.9	0.1	214.882
2	100	**8**	100	200	0.9	0.1	480.187
3	100	**10**	100	200	0.9	0.1	500.689

Acknowledgments This work is supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-(C1090-1011-0001)).Corresponding author is Eui-Nam Huh.

References

1. Weiss A (2007) Computing in the clouds. netWorker 11(4):16–25
2. Interoperability (2009) A key challenge for cloud computing. http://www.lightreading.com/document.asp?doc_id=172033
3. Buyya R, Yeo CS, Venugopal S et al (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener Comput Syst 25:599–616
4. Yamazaki Y (2004) Dynamic collaboration: the model of new business that quickly responds to changes in the market through “The integrated IT/Network Solutions” provided by NEC. NEC Journal of Advanced Technology 1(1):9–16
5. Bubendorfer, K. (2006) Fine grained resource reservation in open grid economies. In: Proceedings of the second IEEE international conference on e-science and grid computing, vol. 1, pp. 81–81
6. Bubendorfer, K, Thomson, W. (2006) Resource management using untrusted auctioneers in a grid economy. In: Proceedings of the second IEEE international conference on e-science and grid computing, vol. 1, pp. 74–74
7. Das A, Grosu D (2005) Combinatorial auction-based protocol for resource allocation in grid. In: Proceedings of the 19th IEEE international parallel and distributed processing symposium
8. Nepal, S., Zic, J. (2008) A conflict neighboring negotiation algorithm for resource services in dynamic collaboration. In: Proceedings of the IEEE international conference on services computing, vol. 2, pp. 7–11
9. Nepal, S., Zic, J., Chan, J. (2007) A distributed approach for negotiating resource contributions in dynamic collaboration. In: Proceedings of the 8th IEEE international conference on parallel and distributed computing applications and technologies, vol. 1, pp. 82–86
10. Ko CS, Kim T, Hwang H (2001) External partner selection using tabu search heuristics in distributed manufacturing. Int J Prod Res 39(17):3959–3974
11. Wu NQ, Su P (2005) Selection of partners in virtual enterprise paradigm. Robot Comput-Integr Manuf 21(5):119–131
12. Wang Z-J, Xu X-F et al (2009) Genetic algorithms for collaboration cost optimization-oriented partner selection in virtual enterprises. Int J Prod Res 47(4):859–881
13. Buyukozkan G, Feyzioglu O, Nebol E (2008) Selection of the strategic alliance partner in logistics value chain. Int J Prod Econ 113:148–158
14. Ip WH, Huang M, Yung KL et al (2003) Genetic algorithm solution for a risk-based partner selection problem in a virtual enterprise. Comput Oper Res 30:213–231

15. Fuqing Z, Yi H, Dongmei Y (2008) A multi-objective optimization model of the partner selection problem in a virtual enterprise and its solution with genetic algorithms. Int J Adv Manuf Technol 37:1220
16. Cheng F, Ye F, Yang J (2009) Multi-objective optimization of collaborative manufacturing chain with time-sequence constraints. Int J Adv Manuf Technol 40:1024–1032
17. Chen YL, Cheng LC, Chuang CN (2008) A group recommendation system with consideration of interactions among group members. Expert Syst Appl 34:2082–2090
18. Sha DY, Che ZH (2005) Virtual integration with a multi-criteria partner selection model for the multi-echelon manufacturing system. Int J Adv Manuf Technol 25:793–802
19. Amid A, Ghodsypour SH, Brien CO (2006) Fuzzy multi-objective linear model for supplier selection in a supply chain. Int J Prod Econ 104:394–407
20. Chang SL, Wang RC et al (2006) Applying fuzzy linguistic quantifier to select supply chain partners at different phases of product life cycle. Int J Prod Econ 100:348–359
21. Chen HH, Lee AHI, Tong Y (2007) Prioritization and operations NPD mix in a network with strategic partners under uncertainty. Expert Syst Appl 33:337–346
22. Saen RF (2007) Supplier selection in the presence of both cardinal and ordinal data. Eur J Oper Res 183:741–747
23. Huang XG, Wong YS, Wang JG (2004) A two-stage manufacturing partner selection framework for virtual enterprises. Int J Comput Integr Manuf 17(4):294–304
24. Gupta P, Nagi R (1995) Optimal partner selection for virtual enterprises in agile manufacturing. http://www.acsu.buffalo.edu/nagi/pubs/pgupta2.ps
25. Fischer M, Jahn H, Teich T (2004) Optimizing the selection of partners in production networks. Robot Comput-Integr Manuf 20(5):593–601
26. Kaya M (2009) MOGAMOD: multi-objective genetic algorithm for motif discovery. Expert Syst Appl 36:2
27. Cowan R, Jonard N et al (2007) Bilateral collaboration and the emergence of innovation networks. Manage Sci 53:1051–1067
28. Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
29. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm.TIK Report no. 103, Swiss Federal Institute of Technology
30. Grosu, D., Das, A (2004). Auction-based resource allocation protocols in grids. In: Proceedings of the 16th IASTED international conference on parallel and distributed computing and systems, vol. 1, pp. 20–27
31. Wolski R, Plank JS, Brevik J, Bryan T (2001) Analyzing market-based resource allocation strategies for the computational grid. Int J High Perform Comput Appl 15(3):258–281
32. Chen, S., Nepal, S., Wang, C., Zic. J. (2008) Facilitating dynamic collaborations with eContract services. In: Proceeding of 2008 IEEE international conference on Web services, vol. 1, pp. 521–528
33. Suzuki, K., Yokoo, M. (2003) Secure generalized Vickery auction using homomorphic encryption. In: Proceedings of 7th international conference on financial cryptography, LNCS, Springer, vol. 2742, pp. 239–249
34. Yokoo, M., Suzuki, K. (2002) Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In: Proceedings of the first joint international conference on autonomous agents and multi-agent systems, vol. 1, ACM Press, pp. 112–119
35. Hwang CL, Yoon K (1981) Multiple attribute decision making: methods and applications. Springer, Berlin