mathematics

MDPI

Article

Balancing Privacy Risk and Benefit in Service Selection for Multiprovision Cloud Service Composition

Linyuan Liu [1,*], Haibin Zhu [2] and Shenglei Chen [1]

[1] Department of E-Commerce, Nanjing Audit University, Nanjing 211815, China; shenglei.chen@nau.edu.cn
[2] Collaborative Systems Laboratory, Nipissing University, North Bay, ON P1B8L7, Canada; haibinz@nipissingu.ca
* Correspondence: liulinyuang@nau.edu.cn

Abstract: The popularity of cloud computing has fueled the growth in multiprovision cloud service composition (MPCSC), where each cloud service provider (CSP) can fulfill multiple tasks, i.e., offer multiple services, simultaneously. In the MPCSC, users would rather disclose some private data for more benefits (e.g., personalized services). However, the more private data is released, the more serious the privacy risk faced by users. In particular, the multiservice provision characteristic of MPCSC further exacerbates the privacy risk. Therefore, how to balance the privacy risk and benefit in service selection for MPCSC is a challenging research problem. In this paper, firstly we explore the service selection problem of balancing privacy risk and benefit in MPCSC (SSBM), then we propose an improved Kuhn–Munkres (KM) algorithm solution to the SSBM problem. Furthermore, we conduct a series of simulation experiments to evaluate the proposed approach. The experimental results show that the proposed approach is both efficient and effective for solving the SSBM problem.

Keywords: cloud computing; data security; privacy risk; personalized services; service selection

MSC: 68N30

Citation: Liu, L.; Zhu, H.; Chen, S. Balancing Privacy Risk and Benefit in Service Selection for Multiprovision Cloud Service Composition. *Mathematics* **2022**, *10*, 1675. https://doi.org/10.3390/math10101675

Academic Editors: Ioana Boureanu and Liqun Chen

Received: 11 April 2022
Accepted: 11 May 2022
Published: 13 May 2022

1. Introduction

Cloud computing has been widely employed as a promising paradigm for building complex distributed software systems by composing existing cloud services (CSs) in the form of business processes [1,2]. With the proliferation of cloud, more and more companies are seeking to deploy their applications in the cloud [3]. According to Garner [4], the global cloud services market will exceed 360 billion US dollars in 2022 and maintain rapid growth. At the same time, a large number of cloud service providers (CSPs) with the ability to provide multiple CSs are constantly emerging on the web [5,6]. For example, the leading CSPs such as Microsoft Azure and IBM Cloud can offer nearly 200 types of CSs, and Tencent Cloud can offer more than 300 types of CSs.

To reduce costs, increase application flexibility and avoid vendor lock-in, the requirements of cloud users are usually fulfilled by multiprovision cloud service composition (MPCSC), where multiple CSPs provide services for the composition and a CSP can offer more than a service [6–8]. In MPCSC, the CSPs need user data to deliver services, study user profiles, and provide personalized services. User data is a crucial resource for CSPs, which can help them understand user preferences and behaviors, thereby improving user experience and enhancing service competitiveness [9]. For users, disclosing personal data to CSPs has both benefit and risk [10,11]. In terms of benefits, users can obtain more convenient services, and reduce transaction time and search costs. For instance, JD.com uses customers' personal data to provide them with one-click shopping and personalized product recommendations. In terms of risks, users lose control over their personal data and even suffer from data abuse risks, such as unauthorized data use and sharing, consistent deceptive advertising, spamming, online stalking, etc. [12,13]. In fact, the willingness

and degree to which users disclose personal data to CSPs depends on their assessment of privacy risks and potential benefits [14].

Researchers have recently explored the relationship between users' data disclosure behavior and their privacy concerns. Most of these works explain the concept of the privacy calculus, which focuses on the trade-off between the costs (i.e., privacy risks) and benefits of private data disclosure [11,15,16]. The main premise of the privacy calculus is that despite strong privacy concerns, users will disclose private data if the benefits they gain justify the cost of losing privacy. The privacy calculus perspective has been widely applied in various scenarios, such as e-commerce [11,14], mobile services [17], and healthcare [18], etc.

In a typical MPCSC, the user's requests (i.e., tasks) need to be assigned to multiple candidate CSs, which utilize the collected user data to perform the tasks and provide personalized services. On one hand, users hope that the privacy risks of disclosing data to these CSs are as small as possible, and on the other hand, they also expect the benefits of personalized services provided by these CSs to be as large as possible. Therefore, how to select appropriate CSs for users is one of the most important issues in MPCSC. In this paper, we attempt to provide a service selection mechanism that enables users to select a set of candidate CSs with the greatest overall utility by balancing privacy risk and benefit. Actually, it is very challenging to design such a service selection mechanism in MPCSC. We summarize three major challenges.

Firstly, users have different privacy preferences for different private data, and each CS requires different private data and provides corresponding data usage policies. Thus, the CS's privacy policies may not fully satisfy the user's privacy requirements, resulting in a spectrum of data abuse risks. Moreover, due to the cloud's openness and virtualization, the CSPs are normally dispatched in different locations and present variable trustworthiness [5,19]. Once the users' sensitive data is disclosed to some low-trust CSPs, users will face more serious privacy leakage risk. In order to help users to select a set of candidate CSs with low privacy risk, a proper privacy risk assessment model that can comprehensively consider the user's privacy requirements, the CS's privacy policies, and the CSP's degree of trustworthiness is needed.

Secondly, since different CSs have different needs for the private data of a user, there are also differences in the quality of the personalized services they provide to the user. It is generally believed that the more private data a CS collects, the more accurate the personalized services it provides to the user [20,21]. Therefore, it needs to establish a mapping relationship between the number of disclosed private data and the benefit of personalized services, and to be able to quantitatively evaluate the personalized benefits provided by CSs.

Thirdly, although the multiservice provision of CSPs really extends the capacity of candidate CSs and improves the system utility of cloud service composition (CSC) [2], it also further exacerbates the risk of privacy breaches. When a CSP offers CSs for different tasks in a CSC, it may collect the private data of a user, and use data mining and machine-learning technologies to acquire formerly hidden sensitive information [22,23]. For instance, a CSP can offer a disease prediction service, using a user's genetic information, which does not expose the person's identity. Simultaneously, it can also offer the doctor appointment service through the user's ID and phone number without using other information. However, a combination of these data can unveil the user's identity and disease. To reduce the inference of privacy information by CSPs, it is necessary to limit the number of services provided by each CSP. In a MPCSC, each of the user's tasks needs to select a suitable candidate service for execution, and each CSP can provide candidate services for multiple tasks. Therefore, the service selection issue of balancing privacy risk and benefit in MPCSC is an n-to-1 task assignment problem. Especially when involving the constraints on the number of service offering, finding the optimal solution for such task assignment problem is nontrivial.

Recently, many research works have focused on the problem of service selection in MPCSC. These works consider the QoS correlation between services [2,24,25], the conflict

and cooperation relationship between services [7], and the requirements of location, cost, reliability, security, and so on [8,26]. They mainly select a set of optimal candidate services from the perspective of satisfying QoS and resource constraints, but most of them ignore the users' privacy requirements. Therefore, how to effectively protect user privacy is still a key issue to be overcome in MPCSC service selection.

In the research of privacy-preserving service composition and selection, some useful approaches were proposed to control the usage and disclosure of private data. However, these approaches still have some shortcomings. Firstly, traditional approaches mainly focus on privacy requirements such as the data sensitivity degree [27,28], the purposes of using data [28–30], and the data retention time [27,31,32], but rarely consider the data storage location requirements. In a distributed cloud service collaboration environment, unrestricted data storage locations will lead to serious privacy leakage risks [33]. In addition, these approaches also rarely support quantitative matching of privacy policies. Hence, it is difficult for traditional approaches to accurately evaluate the privacy risks of cloud services. Secondly, most of these works merely regard the service selection from the perspective of privacy risk [34,35], while ignoring the potential benefit of private data disclosure. Although work [11] has studied the issue of e-commerce service selection that balances the tradeoffs of privacy cost and personalized benefit, they only considered single service selection. Actually, the composite service selection has higher complexity than the single service selection. Moreover, the benefit model in [11] does not consider the correlation between the quality of personalized service and the number of private data. Thirdly, few of these works consider the multiservice provision characteristic of MPCSC.

In our previous work [6], the privacy-regulation-aware service selection problem for MPCSC has been considered. However, although it can support user privacy requirement modeling and privacy policy matching in MPCSC, it can only qualitatively determine whether privacy policies match privacy requirements, and cannot quantitatively calculate the degree of dissatisfaction between them. Second, it lacks a mechanism to quantify the privacy risks and personalized benefits of CSs. Third, its goal is to select a set of candidate CSs with minimal privacy disclosure cost for users. Hence, it is still insufficient to address the service selection issue that balances privacy risk and benefit.

In this work, we quantitatively evaluate the privacy risks and personalized benefits of the CSs and explore the service selection problem that balances privacy risk and benefit in MPCSC (SSBM). The SSBM problem aims to select suitable candidate CSs for multiple tasks by balancing privacy risks and benefits, so that the overall utility of the selected CSs can be maximized while the user's privacy requirements can be satisfied. Such a problem is quite different from the traditional service selection problem, because it comprehensively considers the privacy risks of the user, the personalized benefits provided by the CSs, and the multiservice provision characteristics of CSPs. Therefore, existing approaches are not applicable to this new problem. As far as we know, this work is the first to study the SSBM problem.

To address the SSBM problem, we propose a service selection approach that balances privacy risk and benefit in MPCSC. Specifically, we first quantitatively evaluate the privacy risks and personalized benefits of the CSs. Then, we formulate the SSBM problem as an optimization problem with multiple privacy constraints. Furthermore, since the SSBM problem is an n-to-1 task assignment problem with constraints on the number of service offerings and has high complexity, we propose an improved KM algorithm [36,37] solution to solve the problem. The main contributions of this paper are as follows:

(1) An integer programming optimization model is used to formulate the SSMB problem, which takes into account the privacy risks of the user, the personalized benefits provided by the CSs, and the multiservice provision characteristics of CSPs.
(2) A privacy risk model is proposed to measure the dissatisfaction degree between a CS's privacy policies and the user's privacy preferences, and the CS's privacy risk is evaluated by combining privacy policy dissatisfaction degree, private data sensitivity degree, and CSP's trust degree.

(3) A benefit model is put forward to measure the benefit of personalized service provided by CSs, which employs sigmoid function to model the nonlinear relationship between the quality of personalized service and the number of private data required.
(4) A solution using the improved KM algorithm is designed to solve the SSBM problem. The experimental results demonstrate that the proposed approach can significantly improve the risk–benefit ratio and performance compared with benchmark approaches.

The rest of this paper is structured as follows. Section 2 describes the motivation and scenario of the SSMB problem. Section 3 formally specifies the SSMB problem. Section 4 presents a solution to solve the SSMB problem. The experiments and results are illustrated in Section 5. The related work is reviewed in Section 6. Finally, the conclusion and further works are given in Section 7.

2. Motivation and Scenario

In this section, we demonstrate the relevant characteristics of the SSBM problem with an online pharmacy CSC example. Bob, a hypertensive patient, wishes to purchase daily blood pressure medication through an online pharmacy. The online pharmacy CSC consists of four tasks that are executed in sequence, including *prescription checking*, *medicine ordering*, *payment*, and *shipping* (t_0–t_3). In this example, a total of four CSPs (CSP_0–CSP_3) are involved, each of which can offer candidate CSs for multiple tasks, as shown in Table 1.

Table 1. The CSPs and their candidate CSs.

CSPs	Tasks			
	t_0	t_1	t_2	t_3
CSP_0	CS_{00}	CS_{01}	CS_{02}	Not Available (N/A)
CSP_1	N/A	CS_{11}	CS_{12}	N/A
CSP_2	CS_{20}	N/A	CS_{22}	CS_{23}
CSP_3	CS_{30}	CS_{31}	N/A	CS_{33}

When Bob uses the pharmacy CSC to purchase medicines, he needs to disclose a set of private data to the participating CSs of the pharmacy CSC, e.g., name, ID, address, gender, age, phone number, zip code, bank card number, password, insurance number, prescription, medicines, diagnosis results, allergic history, medical history, medication history, occupation, salary, delivery location, delivery time, and so on. Bob finds that each candidate CS requires not only a set of necessary private data to complete the task function, but also a set of unnecessary private data to provide some personalized services. Meanwhile, Bob also finds that for different candidate CSs that perform the same task, they may require different private data. The private data required by each candidate CS and the personalized services provided are shown in Table 2.

To better illustrate the scenario, Bob presents several reasonable requirements for selecting candidate CSs. First, in order to protect data privacy, Bob expects to select a set of candidate CSs with the lowest possible privacy disclosure risks to perform the tasks. Second, in order to obtain more high-quality personalized services, Bob hopes to select a set of candidate CSs that can provide the highest possible personalized benefits by sacrificing some private data. Third, in order to reduce the aggregation and inference of private data by the CSPs, Bob also wants to limit the maximum number of services that the CSP can provide simultaneously.

In order to effectively solve the problem of service selection that balances privacy risk and benefit, Bob requests a cloud service broker, CSB, to complete this task. After analyzing Bob's request, the CSB learns that the service selection problem is an *n*-to-1 task assignment problem and can be quickly solved by the KM algorithm [36,37]. However, the CSB encounters some challenges in applying the KM algorithm to assign tasks. The first and most important challenge is how to quantify the privacy risk of a CS, which is highly

dependent on the privacy requirements of the user and the privacy protection capability of the CSP. Another challenge is how to quantify the personalized benefit provided by a CS, which depends on the number of private data required by the CS. To this end, we provide the following solutions.

Table 2. The input data and personalized services of the candidate CSs.

Candidate Services	Inputs		Personalized Services
	Necessary Private Data	Unnecessary Private Data	
CS_{00}	**prescription**	**allergies history, medication history**	personalized medicine
CS_{01}	**name**, **address**, **phone number**, **medicines**, checking results	**age, gender, occupation, salary**	medicine recommendation
CS_{02}	**bank card number**, **password**, reservation code	**name**, **phone number**	personalized payment
CS_{11}	**insurance number**, **phone number**, **medicines**, checking results	**medical history**, **medication history**	medicine recommendation
CS_{12}	**name**, **ID**, **phone number**, **bank card number**, reservation code	**address**, **zip code**	personalized payment
CS_{20}	**prescription**	**diagnosis results**	personalized medicine
CS_{22}	**bank card number**, **password**, reservation code	**phone number**	personalized payment
CS_{23}	**name**, **phone number**, **zip code**, payment code	**delivery location**	personalized delivery
CS_{30}	**prescription**	**allergies history**, **medical history**, **medication history**, **diagnosis results**	personalized medicine
CS_{31}	**ID**, **phone number**, **address**, **medicines**, checking results	**age**, **gender**, **medical history**	medicine recommendation
CS_{33}	**name**, **gender**, **address**, **phone number**, payment code	**delivery time**	personalized delivery

The prerequisite for quantifying the CS's privacy risk is to choose the risk measurement indicators. For this reason, through relevant research and analysis [27,35], we empirically divided the measurement indicators of privacy risk into three main criteria: the sensitivity degree of the data, the trust degree of the CSP, and the degree of the user's dissatisfaction regarding the gap between the CS's privacy policies and the user's privacy preferences. Specifically, the user specifies a set of privacy preferences for different private data, such as the sensitivity degree of the data, the purposes of using the data, the storage locations of the data, and the retention time of the data. Correspondingly, each CSP also provides a set of data usage policies for its CSs. When the CS's data usage policies cannot satisfy the user's privacy preferences, the CS's provider may illegally use the user's private data, resulting in serious privacy risks. The greater the degree of dissatisfaction, the greater the privacy risk [27]. In addition, because each piece of private data has a different sensitivity degree and each CSP also has a different trust degree, the disclosure of more highly sensitive private data to CSs provided by low-trust CSPs will cause users to face more serious privacy risks [6,38]. Hence, we comprehensively measure the above three indicators through a privacy risk assessment process, and quantify the risk value and normalize it to a range of 0 to 1, where 0 means no risk at all, while 1 means the highest risk. For example, as shown in Table 3, the trust degrees of CSP_0 and CSP_3 are 0.4 and 0.8, respectively. Although CS_{30} requires more sensitive data and greater privacy dissatisfaction than CS_{00}, its privacy risk may still be lower than CS_{00}. Hence, Bob is willing to assign the *order* tasks to CS_{30}.

Table 3. The CSPs and their trust degrees.

CSPs	CSP_0	CSP_1	CSP_2	CSP_3
Trust degrees	0.4	0.7	0.5	0.8

To quantify the personalized benefit provided by a CS, we model the correlation between the quality of personalized service and the number of private data required. In general, the more private data a CS requests from the user, the easier it is for the provider to establish a complete user profile, which can improve the accuracy of personalized services [20,21]. To this end, according to the number of private data required, we use a benefit function to quantify the benefit value to a range of 0 to 1, and 0 means no benefit at all, while 1 means maximum benefit. For example, because CS_{01} requires more private data than CS_{11}, CS_{01} can provide more benefits than CS_{11}. Therefore, Bob is more willing to assign the *order* tasks to CS_{01}.

Finally, in order not to lose generality, we will illustrate the detailed evaluation process for privacy risk and personalized benefit in Section 3, and conduct large-scale experiments in Section 5.

Based on the pharmacy CSC example, the service selection architecture that balances privacy risk and benefit for an MPCSC is shown in Figure 1.

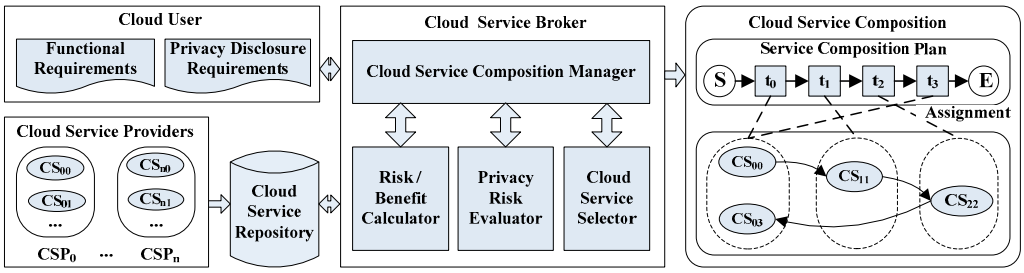


Figure 1. The CS selection architecture that balances privacy risk and benefit for an MPCSC.

In Figure 1, the cloud service broker serves as a mediator between CSPs and cloud users. It is in charge of discovering and selecting candidate CSs provided by different CSPs, and generating an optimal CSC that satisfies the user's requirements including functional and nonfunctional. The nonfunctional ones typically comprise a set of quality-of-service (QoS) parameters, e.g., reliability, availability, response time, price, and reputation, etc. Due to that we are mainly concerned about the risk and benefit of private data disclosure in this paper, we omit other QoS parameters for simplicity. However, the proposed architecture and model can be applied to service selection problems with different QoS parameters.

The core components of the cloud broker include:

(1) Cloud service composition manager: it firstly receives the user's requirements about functions and privacy disclosure, and establishes a service composition plan that describes tasks and their relationships. Then, it searches for candidate CSs for each task from the cloud service repository. Furthermore, it coordinates a risk/benefit calculator, privacy risk evaluator, and cloud service selector to assign suitable candidate CSs to multiple tasks. Finally, it generates an optimal service composition solution.

(2) Risk/benefit calculator: It calculates the privacy risk and benefit of each candidate CS based on the privacy sensitivity preferences of the user, the private data disclosure information of the candidate CSs, and the trust degrees of the CSPs.

(3) Privacy risk evaluator: Based on the privacy risk information of the candidate CSs, it evaluates whether each candidate CS satisfies the privacy risk threshold constraint.

(4) Cloud service selector: According to the results of the privacy evaluation, the privacy risk and benefit information of the candidate CSs, the upper bound of service provision, and the privacy risk and benefit balance weight, the cloud service selector is

in charge of solving the SSMB problem by assigning the suitable candidate CSs to multiple tasks.

3. Problem Model

In this section, we first focus on defining the understanding of the privacy disclosure requirements, MPCSC model, privacy risk model, and personalized benefit model by a set of formal descriptions, and then clearly demonstrate a service selection problem that balances privacy risks and personalized benefits is in MPCSC.

3.1. Privacy Disclosure Requirements

According to the provisions of the General Data Protection Regulation (GDPR) [33], data consumers can only collect private data for legal purposes. At the same time, the GDPR also requires data consumers not to use the collected data for other purposes, and the storage location and retention time of the data must be consistent with those necessary for the stated purpose. To comply with GDPR, each user specifies a set of privacy preferences for their private data, e.g., the sensitivity of the data, the purpose of using the data, the storage location of the data, and the retention time of the data. Correspondingly, each CSP also provides a set of privacy policies for their CSs. The users expect CS's privacy policies to best satisfy their privacy preferences.

In addition, the users not only want to reduce the risks of privacy policy violation as much as possible, but also want to obtain as many the benefits of as many personalized services as possible, and different users have different preferences for the balance of privacy risk and benefit. Furthermore, the users do not want to disclose too much private data to a service, even if the service can provide particularly many benefits. Finally, in order to reduce the collection of private data by each CSP, the users hope to select as few candidate CSs as possible from the same CSP. In summary, the privacy disclosure requirements of the users can be represented as follows:

(1) Private data set PD: $PD = \{pd_0, pd_1, \ldots, pd_{p-1}\}$ expresses a set of private data of the user, where pd_l is the lth private data item, $l \in \{0, 1, \ldots, p-1\}$, p ($= |PD|$) represents the number of private data in PD.

(2) Privacy sensitivity preferences SP: $SP = \{<pd_0, sd_0>, <pd_1, sd_1>, \ldots, <pd_{p-1}, sd_{p-1}>\}$ expresses a set of privacy sensitivity preferences specified by a user, where $<pd_l, sd_l>$ is the lth privacy sensitivity preference, $sd_l \in [0, 1]$ represents the sensitivity degree of pd_l, 0 indicates insensitive, and 1 indicates particularly sensitive.

(3) Privacy disclosure preferences $Pre = \{pre_0, pre_1, \ldots, pre_{p-1}\}$ is a set of privacy preferences specified by a user. Each privacy preference pre_l is defined as a tuple $<pd_l, Pu_l, Lc_l, re_l>$, where $pd_l \in PD$ is a private data item of the user, Pu_l specifies a set of purposes for which the pd_l can be used, Lc_l specifies a set of locations where the pd_l can be stored, $re_g \in N$ specifies the longest time that the CSPs can retain pd_l (in days).

(4) Balance weight w: $w \in [0, 1]$ is a weight parameter that expresses the user's preference for balancing privacy risk and benefit, $w = 0$ means that the user is only concerned about the privacy risk, while $w = 1$ means that the user is only concerned about the benefit, and $w \in (0, 1)$ means that the user considers the trade-off between privacy risk and benefit.

(5) Privacy risk threshold rt: $rt \in 0 \cup R^+$ specifies the maximum privacy risk degree of each service that a user can tolerate.

(6) The upper bound of service provision ub: $ub \in N^+$ is the number of services for a CSP to provide at most in a CSC. For example, $ub = 1$ informs that the CSP can deliver at most one service in a CSC, and $ub > 1$ means that the CSP can offer multiple services in a CSC.

Example 1. *The privacy disclosure requirements of a user for the pharmacy CSC are as follows:*

*PD = {**name**, **ID**, **address**, **gender**, **age**, **phone number**, **zip code**, **bank card number**, **password**, **insurance number**, **prescription**, **medicines**, **diagnosis results**, **allergic history**, **medical history**, **medication history**, **occupation**, **salary**, **delivery location**, **delivery time**},*
*SP = {<**name**, 0.6>, <**ID**, 0.8>, <**address**, 0.6>, <**gender**, 0.5>, <**age**, 0.5>, <**phone number**, 0.7>, <**zip code**, 0.5>, <**bank card number**, 0.75>, <**password**, 0.75>, <**insurance number**, 0.7>, <**prescription**, 0.7>, <**medicines**, 0.7>, <**diagnosis results**, 0.7>, <**allergic history**, 0.65>, <**medical history**, 0.65>, <**medication history**, 0.65>, <**occupation**, 0.4>, <**salary**, 0.4>, <**delivery location**, 0.5>, <**delivery time**, 0.6>},*
*Pre = {pre_0 = <**name**, {ordering, payment, shipping, contact, audit}, {DE, FR, UK, US, CN, AU}, 60>,*

*pre_1 = <**ID**, {ordering, payment}, {DE, FR, UK}, 30>, pre_2 = <**address**, {ordering, payment, shipping, contact}, {DE, FR, UK}, 30>,*
*pre_3 = <**gender**, {shipping, medicine recommendations}, {DE, FR, UK, US, CN, AU}, 60>,*
*pre_4 = <**age**, {medicine recommendations}, {DE, FR, UK, US, CN, AU}, 60>,*
*pre_5 = <**phone number**, {ordering, payment, shipping, contact, audit}, {DE, FR, UK}, 30>,*
*pre_6 = <**zip code**, {shipping, contact, audit}, {DE, FR, UK, US, CN, AU}, 60>, pre_7 = <**bank card number**, {payment}, {DE, FR, UK}, 30>,*
*pre_8 = <**password**, {payment}, {DE, FR, UK}, 30>,*
*pre_9 = <**insurance number**, {ordering}, {DE, FR, UK}, 30>,*
*pre_{10} = <**prescription**, {prescription checking, audit}, {DE, FR, UK}, 30>,*
*pre_{11} = <**medicines**, {ordering}, {DE, FR, UK}, 30>,*
*pre_{12} = <**diagnosis results**, {personalized medicine}, {DE, FR, UK}, 30>,*
*pre_{13} = <**allergic history**, {personalized medicine}, {DE, FR, UK}, 30>,*
*pre_{14} = <**medical history**, {medicine recommendations,personalized medicine}, {DE, FR, UK}, 30>,*
*pre_{15} = <**medication history**, {medicine recommendations, personalized medicine}, {DE, FR, UK}, 30>,*
*pre_{16} = <**occupation**, {medicine recommendations, personalized payment}, {DE, FR, UK, US, CN, AU}, 90>,*
*pre_{17} = <**salary**, {medicine recommendations, personalized payment}, {DE, FR, UK, US, CN, AU}, 90>,*
*pre_{18} = <**delivery location**, {personalized delivery}, {DE, FR, UK, US, CN, AU}, 90>,*
*pre_{19} = <**delivery time**, {personalized delivery}, {DE, FR, UK, US, CN, AU}, 90>},*

w = 0.5,
rt = 2,
ub = 2.

3.2. MPCSC Model

A typical MPCSC includes multiple tasks and multiple CSPs, each CSP has a certain trust degree and can provide candidate CSs for multiple tasks simultaneously.

Let m be the number of CSPs, n be the number of tasks, and o be the number of services provided by a CSP. The set of cloud service providers are expressed as CSPs = {CSP_0, CSP_1, ... , CSP_{m-1}}, where CSP_i is the ith cloud service provider, $i \in \{0, 1, \ldots, m-1\}$. Each CSP_i has a certain trust degree td_i, where $td_i \in [0, 1]$, 0 means completely untrusted, and 1 means completely trusted, the higher the value of td_i, the stronger the privacy protection provided by the CSP_i.

In a cloud service composition environment, users are willing to deliver their tasks and sensitive data to CSs, in reliance on the trust relationship established between users and CSPs. In this paper, the trust degree of the CSPs is a key indicator for assessing the privacy risk of CSs. The calculation of trust degree needs to consider the behaviors of CSPs, such as security-related behaviors, QoS-related behaviors, etc.

At present, many research efforts have proposed trust computing schemes for CSPs [19,39,40]. As the trust computing scheme proposed in [40] has the advantages of high speed, low overhead, and consideration of security-related behaviors, this paper adopts this scheme to evaluate the trust degree of the CSPs. The trustworthiness computing process of the scheme mainly consists of two phases: (1) It utilizes a set of distributed monitoring agents to quickly perceive the trusted behavior of VMs in cloud environments. Specifically, these agents monitor and collect the security-related and QoS-related behavior of virtual machines. Security-related behavior indicators include authentication mechanisms, authorization mechanisms, security protection mechanisms, and the number of illegal access or scanning of sensitive ports, etc. QoS-related behavior indicators include current CPU usage, memory usage, hard disk usage, average response time, and average task success rate, etc. (2) Based on large-scale, real-time, and multidimensional behavior data perceived by the distributed agents, it uses a combination of time window and time decay function to compute the trustworthiness of VMs, which can effectively satisfy the accuracy requirement of trustworthiness computing. To save space, please refer to [40] for the detailed computing process.

The set of cloud services provided by CSP_i is expressed as $CSs_i = \{CS_{i0}, CS_{i2}, \ldots, CS_{io-1}\}$, where CS_{ik} is the kth CS provided by CSP_i, $k \in \{0, 1, \ldots, o-1\}$. Each CS_{ik} includes a set of inputs IN_{ik}, a set of outputs OUT_{ik}, a set of functions FUN_{ik}, and provides a set of privacy policies $Pol_{ik} = \{pol_{ik}^0, pol_{ik}^1, \ldots, pol_{ik}^{q-1}\}$, where q ($= |PreR|$) expresses the size of the privacy policies Pol_{ik}. Each privacy policy pol_{ik}^h is defined as a tuple $<pd_{ik}^h, Pu_{ik}^h, lc_{ik}^h, re_{ik}^h>$, $h \in \{0, 1, \ldots, q-1\}$, where $pd_{ik}^h \in IN_{ik}$ is a private data item for which the policy is defined, Pu_{ik}^h is a set of purposes for CS_{ik} using pd_{ik}^h, lc_{ik}^h is the location where the CS_{ik}stores pd_{ik}^h, re_{ik}^h is the time for CSP_i to retain pd_{ik}^h (in days).

The tasks and their relationships in MPCSC are usually described by a cloud service composition plan. Formally, it is represented as a directed acyclic graph $G = (T, E)$, nodes $T = \{t_0, t_1, \ldots, t_{n-1}\}$ represent a set of tasks where t_j is the jth task, $j \in \{0, 1, \ldots, n-1\}$, and edges $E = \{(t_e, t_g) \mid t_e, t_g \in T\}$ are a set of links between tasks, which represent data and task dependencies. More specifically, each task t_j in graph G has a set of candidate CSs provided by different CSPs, and a suitable candidate CS needs to be selected to fulfill its function.

3.3. Privacy Risk Model

It is usually difficult to fully satisfy the user's privacy preferences in the privacy policies of a CS. Therefore, the CS's provider may use private data without authorization, resulting in serious privacy risks. Given CS_{ik}'s privacy policies $Pol_{ik} = \{pol_{ik}^0, pol_{ik}^1, \ldots, pol_{ik}^{q-1}\}$ and the user's privacy preferences $Pre = \{pre_0, pre_1, \ldots, pre_{p-1}\}$, we first measure the dissatisfaction degree between each privacy policy of CS_{ik} and the corresponding privacy preference. More specifically, we calculate the dissatisfaction degree corresponding to the purpose, location, and retention privacy attributes separately, as described in the following:

(1) Purpose dissatisfaction: The purpose attributes of both privacy policies and privacy preferences are defined as a purpose set; we use the Jaccard coefficient [41] to measure the distance between them. The degree of dissatisfaction with the purpose attribute is measured by:

$$f_{pu}^d(Pu_{ik}^h, Pu_l) = \begin{cases} 0, & if\ Pu_{ik}^h \subseteq Pu_l \\ 1 - \frac{|Pu_{ik}^h \cap Pu_l|}{|Pu_{ik}^h \cup Pu_l|}, & otherwise \end{cases} \quad (1)$$

(2) Location dissatisfaction: The privacy preference specifies a set of locations where a private data can be stored. We measure the degree of dissatisfaction with the location by judging whether the location attribute of the privacy policy is in the location set specified by the privacy preference, which is calculated by:

$$f_{lc}^{d}(lc_{ik}^{h},\ Lc_{l}) = \begin{cases} 0, & if\ lc_{ik}^{h} \in\ Lc_{l} \\ 1, & otherwise \end{cases} \tag{2}$$

(3) Retention dissatisfaction: The retention can be expressed as a numerical value. We measure the degree of dissatisfaction with the retention by evaluating whether the retention time of the privacy policy is less than or equal to the retention time of the privacy preference, which is calculated by:

$$f_{re}^{d}(re_{ik}^{h}, re_{l}) = \begin{cases} 0, & if\ re_{ik}^{h} \leq re_{l} \\ 1, & otherwise \end{cases} \tag{3}$$

As can be seen from (2) and (3), the degree of dissatisfaction with the location and retention attributes takes values of 0 or 1 (satisfied or dissatisfied). If the measurement result of any one of them is 1, it may cause the whole privacy policy to fail to satisfy the corresponding privacy preference. Hence, the degree of dissatisfaction with the gap between the privacy policy pol_{ik}^{h} and the corresponding privacy preference pre_{l} is measured by:

$$f_{pol}^{d}(pol_{ik}^{h}, pre_{l}) = \begin{cases} 1, & if\ f_{lc}^{d}(lc_{ik}^{h},\ Lc_{l}) = 1\ \vee\ f_{re}^{d}(re_{ik}^{h}, re_{l}) = 1 \\ f_{pu}^{d}(Pu_{ik}^{h}, Pu_{l}), & otherwise \end{cases} \tag{4}$$

where $Pu_{ik}^{h} \in pol_{ik}^{h}$, $lc_{ik}^{h} \in pol_{ik}^{h}$, $re_{ik}^{h} \in pol_{ik}^{h}$, $Pu_{l} \in pre_{l}$, $Lc_{l} \in pre_{l}$, and $re_{l} \in pre_{l}$.

The National Institute of Standards and Technology (NIST) has defined risk as a function of threat probability and impact [42]. In this paper, we adopt the same definition to evaluate the privacy risk of a CS. Specifically, we determine the threat likelihood based on the trust degree of the CSP, and consider the impact based on the sensitivity degree of the disclosed private data and the degree of dissatisfaction with the privacy policy. The more trustworthy the CSP, the lower the privacy risk, and conversely, the more sensitive the data, the higher the privacy risk [6,43]. Meanwhile, the more private data disclosed, the higher the privacy risk [28,38]. The private data disclosed to a service and the corresponding sensitivity degree of these data can be expressed as:

(1) Privacy disclosure vector DV_{ik}: It is a vector of length p, where $DV_{ik}[l] \in [0, 1]$ denotes if the private data item pd_{l} is disclosed to CS_{ik}, $DV_{ik}[l] = 1$ means yes and 0 no, $i \in \{0, 1, \ldots, m-1\}$, $k \in \{0, 1, \ldots, o-1\}$, $l \in \{0, 1, \ldots, p-1\}$. The $DV_{ik}[l]$ is calculated by:

$$DV_{ik}[l] = \begin{cases} 1, & if\ pd_{l} \in IN_{ik} \\ 0, & otherwise \end{cases} \tag{5}$$

(2) Privacy sensitivity vector SV: It is a vector of length p, where $SV[l] = sd_{l}$ denotes the sensitivity degree of the lth private data item pd_{l}, $l \in \{0, 1, \ldots, p-1\}$.

Given DV_{ik}, SV, f_{ik}^{D} and td_{i}, the privacy risk of CS_{ik} provided by CSP_{i} can be evaluated by:

$$f_{ik}^{r} = \sum_{l=0}^{p-1} DV_{ik}[l] \times SV[l] \times f_{pol}^{d}(pol_{ik}^{h}, pre_{l}) \times (1 - td_{i}) \tag{6}$$

where $f_{ik}^{r} \in 0 \cup R^{+}$, $f_{pol}^{d}(pol_{ik}^{h}, pre_{l}) \in f_{ik}^{D}$.

Example 2. *In Table 2, CS_{00} requests to disclose a set of private data of the user, i.e., prescription, allergies history, medication history. CSP_{0}'s trust degree is illustrated in Table 3, i.e., $td_{0} = 0.4$. The user's privacy requirements for these data are shown in Example 1. Assumed that the privacy policies specified by CSP_{0} for these data are as follows:*

pol_{00}^{0} = < ***prescription***, *{prescription checking, audit, medical analysis, advertisement}, US, 90*>
pol_{00}^{1} = < ***allergic history***, *{personalized medicine, medicine recommendations}, DE, 30*>

pol_{00}^{2} = < ***medication history***, *{medicine recommendations, personalized medicine, advertisement}, UK, 30>*

For privacy policy pol_{00}^{0}, its corresponding privacy preference is pre_{10}, the degree of dissatisfaction with the purpose attribute can be calculated as:

$$f_{pu}^{d}\left(Pu_{00}^{0}\,,\ Pu_{10}\right) = 1 - \frac{2}{4} = 0.5$$

Likewise, the degree of dissatisfaction with the location attribute can be calculated as:

$$f_{lc}^{d}\left(lc_{00}^{0},\ Lc_{10}\right) = 1$$

In the same way, the degree of dissatisfaction with the retention attribute can be calculated as:

$$f_{re}^{d}\left(re_{ik}^{h},\ re_{l}\right) = 1$$

Based on the degree of dissatisfaction with the purpose, location and retention attributes, the degree of dissatisfaction with the gap between the policy pol_{00}^{0} and the preference pre_{10} can be calculated as:

$$f_{pol}^{d}\left(pol_{00}^{0},\ pre_{10}\right) = 1$$

Similarly, the degrees of dissatisfaction with pol_{00}^{1} and pol_{00}^{2} can be calculated as:

$$f_{pol}^{d}\left(pol_{00}^{1},\ pre_{13}\right) = 0.5 f_{pol}^{d}\left(pol_{00}^{2},\ pre_{15}\right) = 0.33$$

The privacy disclosure vector of CS_{00} can be expressed as:
DV_{00} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0]
The privacy sensitivity vector for the pharmacy CSC can be expressed as:
SV = [0.6, 0.8, 0.6, 0.5, 0.5, 0.7, 0.5, 0.75, 0.75, 0.7, 0.7, 0.7, 0.7, 0.65, 0.65, 0.65, 0.4, 0.4, 0.5, 0.6]

Based on DV_{00}, SV, $f_{pol}^{d}(pol_{00}^{0}, pre_{10})$, $f_{pol}^{d}(pol_{00}^{1}, pre_{13})$, $f_{pol}^{d}(pol_{00}^{2}, pre_{15})$ and td_0, the privacy risk of CS_{00} is evaluated as follows:

$$\begin{aligned} f_{ik}^{r} &= DV_{00}[10] \times SV[10] \times f_{pol}^{d}(pol_{00}^{0}, pre_{10}) \times (1 - td_0) + DV_{00}[13] \times SV[13] \times f_{pol}^{d}(pol_{00}^{0}, pre_{13}) \\ &\times (1 - td_0) + DV_{00}[15] \times SV[15] \times f_{pol}^{d}(pol_{00}^{2}, pre_{15}) \times (1 - \mathrm{td}_0) \\ &= 1 \times 0.7 \times 1 \times 0.6 + 1 \times 0.65 \times 0.5 \times 0.6 + 1 \times 0.65 \times 0.33 \times 0.6 \\ &= 0.743 \end{aligned}$$

3.4. Personalized Benefit Model

The more user private data are requested by CSs, the more benefits of personalized service they provide [18,19]. When the amount of private data required by a CS is relatively small, it is difficult for the provider to build the user profile based on these data, resulting in low accuracy of the personalized services provided. As the number of private data required gradually increases, the accuracy of personalized services is rapidly increasing. However, when the required private data is close to the upper limit of the private data disclosed by the user in a CSC, the CSP can establish the user's basic profile, so the accuracy of personalized services has gradually slowed down. In short, with the number of private data required continues to increase, the benefits that the user obtains from personalized services initially increase slowly, then accelerate, and finally converge, that is, there is a nonlinear correlation between the benefit and the number of required private data.

Recently, several works use the sigmoid function to measure the correlation between the quality of experience (QoE) and QoS [44,45]. For this reason, we introduce a sigmoid function, and make some appropriate translation and scaling transformation to make it

show the correlation between the quality of the personalized service and the amount of private data required.

The benefit of personalized service provided by CS_{ik} is related to the amount of private data it requests, which can be measured by:

$$f_{ik}^{b} = \frac{1}{1 + e^{-\alpha(npd_{ik} - \beta)}} \tag{7}$$

where $f_{ik}^{b} \in (0, 1)$, $i \in \{0, 1, \ldots, m-1\}$, $k \in \{0, 1, \ldots, o-1\}$, $npd_{ik} = \sum_{l=0}^{p} DV_{ik}[l]$ is the number of private data required by the CS_{ik}, α and β are two domain-specific parameters, which control the growth rate and the midpoint of the benefit function respectively.

Assuming that *PDS* is a set of all private data requested by a CSC, it is generally specified by the developer of the CSC, e.g., a cloud service broker. Each service collects only a part of the user's private data, i.e., $npd \leq |PDS|$, $|PDS| \geq 0$. Since the values of the original sigmoid function on the *X*-axis are a set of real numbers where the midpoint is at 0, the midpoint of the benefit function needs to be translated from 0 to the right along the *X*-axis by $|PDS|/2$ units, i.e., $\beta = |PDS|/2$. For example, in the pharmacy CSC, if the number of the user's private data values is 20, then $\beta = 10$. The α parameter characterizes the growth rate of the benefit function, and it cannot be set too small or too large. If it is too small, the growth of the benefit value will be too gentle, and conversely, the growth of the benefit value will be too steep. Hence, we will set α according to the specific application scenario. For example, in Figure 2, the curve of $\alpha = 0.1$ grows too gently, while the curve of $\alpha = 1$ grows too steeply. For the pharmacy CSC, it is reasonable to set α to 0.5, as shown by the curve of $\alpha = 0.5$.

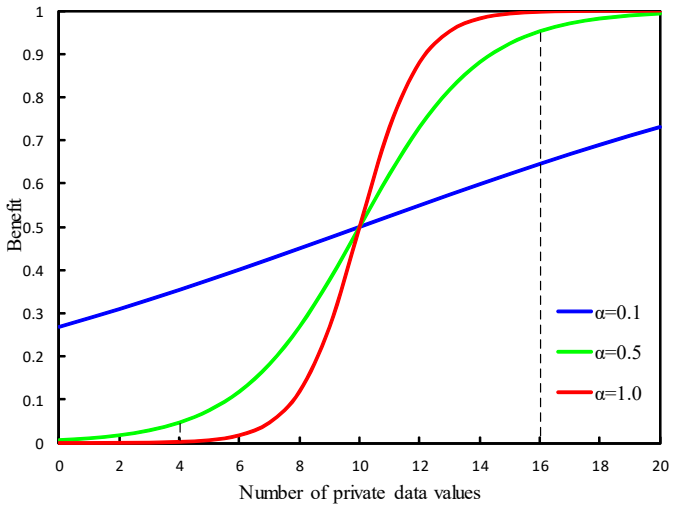


Figure 2. The correlation curves between the benefit and the number of private data values required.

Example 3. *In Figure 2, β is set to 10, the curve of $\alpha = 0.5$ start to increase slowly, then accelerate, and finally converge. For example, when $npd = 4$, the benefit value corresponding to the curve $\alpha = 0.5$ is 0.0474. However, when $npd = 16$, the benefit value corresponding to the curve $\alpha = 0.5$ increases to 0.9525.*

3.5. Problem Definition

Based on the MPCSC model, privacy disclosure requirements, privacy risk model, and benefit model, we will formally define the SSBM problem below. To illustrative the SSMB problem, some specific data structures can be formalized as follows:

(1) Provision matrix P: It is an $m \times n$ matrix, where $P[i, j]$ denotes whether there is a CS_{ik} in CSP_i that can execute the task t_j, $P[i, j] = CS_{ik}$ means CS_{ik} is a candidate service of t_j and 0 no, $i \in \{0, 1, \ldots, m-1\}$, $j \in \{0, 1, \ldots, n-1\}$, $k \in \{0, 1, \ldots, o-1\}$.

(2) Risk matrix R: It is an $m \times n$ matrix, where $R[i, j] \in [0, 1]$ denotes the normalized privacy risk of CS_{ik} provided by CSP_i for task t_j, $i \in \{0, 1, \ldots, m-1\}$, $j \in \{0, 1, \ldots, n-1\}$, $k \in \{0, 1, \ldots, o-1\}$. $R[i, j]$ is calculated by:

$$R[i,j] = \begin{cases} \frac{f_{ik}^r - \min\{R\}}{\max\{R\} - \min\{R\}} & if\ P[i,\ j] = CS_{ik} \\ 0 & otherwise \end{cases} \tag{8}$$

where $\max\{R\}$ and $\min\{R\}$ express the maximum and minimum values of privacy risks in all candidate CSs, respectively. Because the privacy risk value of the CS is a non-negative real number, in order to make it comparable to the value of the benefit, we use the Min–Max normalization to uniformly scale it to the range [0, 1].

(3) Benefit matrix B: It is an $m \times n$ matrix, where $B[i, j] \in (0, 1)$ denotes the benefit of CS_{ik} provided by CSP_i for task t_j, $i \in \{0, 1, \ldots, m-1\}$, $j \in \{0, 1, \ldots, n-1\}$, $k \in \{0, 1, \ldots, o-1\}$. $B[i, j]$ is obtained by:

$$B[i,j] = \begin{cases} f_{ik}^b & if\ P[i,j] = CS_{ik} \\ 0 & otherwise \end{cases} \tag{9}$$

(4) Evaluation matrix E: It is an $m \times n$ matrix, where $E[i, j]$ denotes whether CS_{ik} provided by CSP_i for task t_j meets the user's privacy risk threshold constraint, $E[i, j] = 1$ means yes and 0 no, $i \in \{0, 1, \ldots, m-1\}$, $j \in \{0, 1, \ldots, n-1\}$, $k \in \{0, 1, \ldots, o-1\}$. $E[i, j]$ is obtained by:

$$E[i,j] = \begin{cases} 1, & if\ P[i,j] = CS_{ik} \wedge f_{ik}^r \le rt \\ 0, & otherwise \end{cases} \tag{10}$$

Example 4. *Figure 3 shows the provision, risk, benefit, and evaluation matrixes of the pharmacy CSC. Figure 3a is a 4 × 4 matrix, where the rows express CSP_0–CSP_3, and the columns express t_0–t_3, and the matrix element CS_{ik} represents that it is a candidate service provided by CSP_i for task t_j. According to Figure 3a, the risk, benefit, and evaluation matrixes are calculated by (8), (9), and (10), respectively, as shown in Figure 3b–d. The underlined data in Figure 3d represents candidate CSs that do not meet the privacy risk threshold constraints.*

$$\begin{bmatrix} CS_{00} & CS_{01} & CS_{02} & 0 \\ 0 & CS_{11} & CS_{12} & 0 \\ CS_{20} & 0 & CS_{22} & CS_{23} \\ CS_{30} & CS_{31} & 0 & CS_{33} \end{bmatrix} \quad \begin{bmatrix} 0 & 0.24 & 0.09 & 0 \\ 0 & 0.83 & 1 & 0 \\ 0.08 & 0 & 0.24 & 0.26 \\ 0.31 & 0.47 & 0 & 0.25 \end{bmatrix}$$

(**a**) (**b**)

$$\begin{bmatrix} 0.06 & 0.31 & 0.08 & 0 \\ 0 & 0.12 & 0.17 & 0 \\ 0.04 & 0 & 0.06 & 0.08 \\ 0.12 & 0.23 & 0 & 0.12 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & \underline{0} & \underline{0} & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

(**c**) (**d**)

Figure 3. The provision, risk, benefit, and evaluation matrixes for the pharmacy CSC. (**a**) Provision matrix; (**b**) risk matrix; (**c**) benefit matrix; (**d**) evaluation matrix.

Based on the risk, benefit, and evaluation matrixes, we can select CSs for MPCSC. The SSBM problem is how to select a CSC with a minimal privacy risk and a maximal benefit from all candidate CSCs to meet the privacy disclosure requirements of the user. Obviously, this is a multiobjective optimization problem (MOOP), in which the objectives of minimizing privacy risk and maximizing benefit often conflict with each other. In this paper, we convert the SSBM problem with dual objectives into a problem of utility maximization to balance the privacy risk and benefit. By maximizing the overall utility of a CSC, the task assignment solution intends to achieve a balance between minimizing the privacy risk and maximizing the benefit. Aiming at the above problem, we formalize the following two data structures:

(1) Utility matrix U: It is an $m \times n$ matrix, where $U[I, j] \in [-1, 1)$ denotes the utility of CS_{ik} provided by CSP_i for task t_j, $i \in \{0, 1, \ldots, m-1\}$, $j \in \{0, 1, \ldots, n-1\}$, $k \in \{0, 1, \ldots, o-1\}$. $U[i, j]$ is calculated by:

$$U[i, j] = w \times B[i, j] - (1 - w) \times R[i, j] \tag{11}$$

(2) Assignment matrix A: It is an $m \times n$ matrix, where $A[i, j] \in \{0, 1\}$ denotes whether t_j is assigned to CS_{ik} ($A[i, j] = 1$) or not ($A[i, j] = 0$), $i \in \{0, 1, \ldots, m-1\}$, $j \in \{0, 1, \ldots, n-1\}$, $k \in \{0, 1, \ldots, o-1\}$.

Given U, E, and ub, the SSBM problem is to find a matrix A to:

$$\text{Max} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} U[i, j] \times A[i, j] \tag{12}$$

subject to:

$$A[i, j] \in \{0, 1\} \quad (i \in \{0, 1, ..., m-1\},\ j \in \{0, 1, ..., n-1\}) \tag{13}$$

$$\sum_{i=0}^{m-1} A[i, j] = 1 \quad (j \in \{0, 1, ..., n-1\}) \tag{14}$$

$$\sum_{j=0}^{n-1} A[i, j] \leq ub \quad (i \in \{0, 1, ..., m-1\}) \tag{15}$$

$$E[i, j] \times A[i, j] > 0 \quad (i \in \{0, 1, ..., m-1\},\ j \in \{0, 1, ..., n-1\}) \tag{16}$$

where Constraint (13) specifies that the decision variables are binary; Constraint (14) guarantees that each task is only assigned to one CS; Constraint (15) ensures that the number of services offered by each CSP does not exceed the upper bound of service provision; and Constraint (16) ensures that the privacy risk of each assigned CS is below the privacy risk threshold.

Example 5. *In Example 1, w, rt, and ub are specified as 0.5, 2, and 2, respectively. Based on the risk, benefit, and evaluation matrixes, the utility matrix of the pharmacy CSC is obtained by (11), as shown in Figure 4a. In Figure 4a, the assignment solution with the maximal utility (−0.005) should be: {CS_{00}, CS_{01}, CS_{02}, CS_{33}}. However, this solution has three CSs (CS_{00}, CS_{01}, and CS_{02}) provided by CSP_0, which should be eliminated due to the upper bound constraint of service provision, e.g., ub = 2. Among the remaining candidate solutions, the assignment solution with the maximal utility (−0.055) is: {CS_{20}, CS_{01}, CS_{02}, CS_{33}}. Finally, the optimal solution that meets the privacy requirements of the user is found. The assignment matrix of the pharmacy CSC is demonstrated in Figure 4b.*

$$\begin{bmatrix} 0.03 & \boxed{0.035} & -0.005 & 0 \\ 0 & -0.355 & -0.415 & 0 \\ -0.02 & 0 & -0.09 & -0.09 \\ -0.095 & -0.12 & 0 & -0.065 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(**a**) (**b**)

Figure 4. The utility and assignment matrix for the pharmacy CSC. (**a**) utility matrix; (**b**) assignment matrix.

4. Solution to the SSBM Problem

The SSBM problem is a typical one-to-many task assignment problem. When the brute force search method is used to solve this problem, the solution space can be up to $O(m^n)$ [46]. Therefore, we can use the well-known KM algorithm that has a complexity of $O(m^3)$ [36,37,47]. However, the KM algorithm can only solve the generalized task assignment problem. For the SSBM problem, the KM algorithm has the following limitations:

(1) The KM algorithm always finds the solution with the smallest sum [48]. However, the SSBM problem requires finding a solution with the greatest utility.
(2) For the SSMB problem, the KM algorithm can always find a result for it, but the result may not be a feasible solution. For example, when the CSP_i cannot provide candidate CS_{ik} for task t_j or the candidate CS_{ik} cannot satisfy the privacy risk threshold constraint, i.e., $E[i, j] = 0$, the KM algorithm may produce incorrect task assignments, leading to an infeasible solution.
(3) The KM algorithm can only solve the 1-to-1 task assignment problem, that is, $m = n$, and a CSP can only serve one task in a CSC. However, the SSMB problem is a typical n-to-1 task assignment problem. In SSMB, $m \gg n$, where "$\gg$" means "much larger than", and each CSP can provide services for multiple tasks of a CSC.

To deal with the above limitations, we propose an improved KM (IKM) algorithm solution to solve the SSMB problem. The basic idea of IKM is: firstly, we build a utility matrix U according to the risk matrix R, benefit matrix B, and balance weight w. Secondly, we reset the corresponding elements in U according to the evaluation matrix E, and then extend the reset U matrix to a square matrix M that can be processed by the KM algorithm. Thirdly, based on M, we use the KM algorithm to solve the SSMB. The IKM solution includes the following three steps:

Step 1: Utility Matrix Building

In this step, we build the utility matrix of candidate CSs based on their risk and benefit matrixes. Concretely, for any CSP_i belonging to CSPs and t_j belonging to T, we first calculate the normalized privacy risk and benefit of CS_{ik} by (8) and (9), respectively, and set them to $R[i, j]$ and $B[i, j]$. Then, we calculate the utility of CS_{ik} by (11) according to $R[i, j]$, $B[i, j]$, and w, and set it to $U[i, j]$. Finally, we return the U matrix. The details of this step are described in Algorithm 1.

Algorithm 1: Utility matrix building

Input: m: the number of CSPs; n: the number of tasks;
w: the privacy risk and benefit balance weight.
Output: U: a utility matrix.
1: **for** $i = 0, 1, \ldots m - 1$ **do**
2: **for** $j = 0, 1, \ldots, n - 1$ **do**
3: $R[i, j] \leftarrow$ calculate the normalized privacy risk of CS_{ik} by (8);
4: $B[i, j] \leftarrow$ calculate the benefit of CS_{ik} by (9);
5: $U[i, j] \leftarrow$ calculate the utility of CS_{ik} from $R[i, j]$, $B[i, j]$ and w by (11);
6: **end for**
7: **end for**
8: return U;

Step 2: Utility Matrix Reset and Extension

In step 1, we have built the U matrix. Because the KM algorithm cannot be directly used to solve the SSMB problem, we need to reset and extend the U matrix in this step. Specifically, firstly, we find the maximum utility $maxu$ in matrix U, e.g., boxed data in Figure 4a. Secondly, for any CSP_i belonging to CSPs and t_j belonging to T, we evaluate whether CS_{ik} satisfies the privacy risk threshold constraint, and set the evaluation result to $E[i, j]$. If CS_{ik} passes the evaluation check, e.g., $E[i, j] = 1$, we reset $U[i, j]$ to $maxu$-$U[i, j]$; otherwise, we reset $U[i, j]$ to n. This is because we need to find a CSC with the maximum utility. Besides this, due to $U[i, j] \in [-1, 1)$, the utility of a CSC never exceeds n, thus we use n as a particularly large value to replace those elements in U that have not passed the privacy evaluation. The reset U matrix of the pharmacy CSC is shown in Figure 5a. Thirdly, we construct an $m \times ub$ rows and n columns matrix M, extend each row in U to ub rows and update them to the corresponding rows in M. Then we determine whether the number of rows in M is greater than the number of columns, i.e., $m \times ub > n$. If the equation holds, we add $m \times ub - n$ virtual columns in M, where each column expresses a virtual task. Because virtual tasks do not need to be assigned services, we set their utility value to 0. Finally, we return the M matrix. The extended U matrix of the pharmacy CSC is shown in Figure 5b, and the details of this step are specified in Algorithm 2.

Algorithm 2: Utility matrix reset and extension

Input: m: the number of CSPs; n: the number of tasks;
ub: the upper bound of service provision; U: the utility matrix.
Output: M: a square matrix extended from U matrix.

$maxu \leftarrow$ finds the maximum utility in matrix U;
for $i = 0, 1, \ldots, m - 1$ **do**
  for $j = 0, 1, \ldots, n - 1$ **do**
    $E[i, j] \leftarrow$ calculate the privacy evaluation result of CS_{ik} by (10);
    if $E[i, j] = 1$ **then**
      $U[i, j] \leftarrow maxu$-$U[i, j]$;
    else
      $U[i, j] \leftarrow n$;
    end if
  end for
end for
for $j = 0, 1, \ldots, n - 1$ **do**
  for $i = 0, 1, \ldots, m - 1$ **do**
    $index \leftarrow 0$;
    for $x = 0, 1, \ldots, ub - 1$ **do**
      $M[index{+}{+}, j] \leftarrow U[i, j]$;
    end for
  end for
end for
if $m \times ub > n$ **then**
  for $z = n, n + 1, \ldots, m \times ub - 1$ **do**
    for $y = 0, 1, \ldots, m \times ub - 1$ **do**
      $M[y, z] \leftarrow 0$;
    end for
  end for
end if
return M;

$$\begin{bmatrix} 0.005 & 0 & 0.04 & 4 \\ 4 & 4 & 4 & 4 \\ 0.055 & 4 & 0.125 & 0.125 \\ 0.13 & 0.155 & 4 & 0.1 \end{bmatrix}$$

(a) reset utility matrix

$$\left[\begin{array}{cccc|cccc} 0.005 & 0 & 0.04 & 4 & 0 & 0 & 0 & 0 \\ 0.005 & 0 & 0.04 & 4 & 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0.055 & 4 & 0.125 & 0.125 & 0 & 0 & 0 & 0 \\ 0.055 & 4 & 0.125 & 0.125 & 0 & 0 & 0 & 0 \\ 0.13 & 0.155 & 4 & 0.1 & 0 & 0 & 0 & 0 \\ 0.13 & 0.155 & 4 & 0.1 & 0 & 0 & 0 & 0 \end{array}\right]$$

(b) extended utility matrix

Figure 5. The reset utility matrix and extended utility matrix.

Step 3: KM Algorithm-Based Task Assignment

In this step, we use the KM algorithm to assign suitable candidate CSs to n tasks according to the M matrix. First of all, we call the KM algorithm to obtain an $m \times ub$ rows and 2 columns temporary matrix N, where each row expresses an assignment. Then, we form the assignment matrix A according to N. Since there may be some incorrect assignments in A, they should be eliminated from the assignment solution. Third, we traverse the A matrix and judge whether the utility corresponding to each assignment is equal to n, i.e., $A[i, j] = 1$ and $U[i, j] = n$. If the judgment conditions are held, we prevent the assignment of CSs to these tasks and return failure. Finally, we determine whether each task is assigned to a candidate CS. If yes, we calculate the overall utility of the CSC by (12) and succeed; otherwise, it is failed. The details of this step are shown in Algorithm 3.

Algorithm 3: Optimal task assignment

Input: M: A square matrix extended from U matrix.
Output: Success: A; Failure: no feasible A is obtained.
1: $N \leftarrow$ **KM** (M);
2: Form the assignment matrix A based on N;
3: **for** $j = 0, 1, \ldots, n-1$ **do**
4: **for** $i = 0, 1, \ldots\ m-1$ **do**
5: **if** $A[i, j] = 1$ and $U[i, j] = n$ **then**
6: **return** Failure;
7: **end if**
8: **end for**
9: **end for**
10: **if** for all columns of matrix A satisfy $\sum_{i=0}^{m} A[i, j] = 1$ **then**
11: calculate the overall utility of the optimal CSC by (12);
12: **return** Success
13: **else**
14: **return** Failure
15: **end if**

The complexity of the above algorithms depends on the number of CSPs (m), the number of tasks (n), and the service provision upper bound (ub). The time complexity of Algorithm 1 is $O\ (m \times n)$. The time complexity of Algorithm 2 is decided by: (1) the time complexity of finding the maximum utility and resetting the U matrix, i.e., $O\ (m \times n) + O\ (m \times n)$, and (2) the time complexity of extending the U matrix, i.e., $O\ (m \times n \times\ ub) + O\ ((m \times ub) \times (m \times ub - n))$. Thus, the overall complexity of Algorithm 2 is $2O\ (m \times n) + O\ ((m \times ub)^2)$. For Algorithm 3, its time complexity is determined by: (1) the time complexity of calling the KM algorithm and forming the assignment matrix, i.e., $O\ ((m \times ub)^3) + O\ (m \times ub)$, and (2) the time complexity of eliminating incorrect task

assignments and calculating CSC utility, i.e., $O(m \times n) + O(n)$. Thus, the overall complexity of Algorithm 3 is $O((m \times ub)^3) + O(m \times ub) + O(m \times n) + O(n)$.

In the presented scenarios, ub is a constant (typically less than 5), and $m >> n$. Consequently, the time complexity of the entire solution can be simplified as: $O((m \times ub)^3) + O((m \times ub)^2) + 4O(m \times n) + O(m \times ub) + O(n) = O(m^3)$.

5. Experiments

Using the hardware and software configuration in Table 4, we make two sets of experiments to assess the efficiency and effectiveness of the IKM approach. As far as we know, there is no other research directly related to our study. Hence, we set the MinR and MaxB as the effectiveness benchmarks and Cplex as the performance benchmark. Since IKM, MinR, and MaxB are all based on the KM algorithm to solve their problems, their efficiency is the same, thus we only compare their effectiveness. In addition, both IKM and Cplex are optimization methods to solve the SSBM problem and have the same result, therefore we only compare their time performance. Note that, we randomly assigned the values of some parameters in the experimental section, such as the number of tasks (n), the number of CSPs (m), the weight of risk and benefit balance (w), the privacy risk threshold (rt), the number of private data required (npd), and the service provision upper bound (ub). Such random choices follow a common simulation process, i.e., all the assigned values are possible in reality.

Table 4. The hardware and software configurations.

Type	Configuration
Environment	Windows 7 Enterprise (64-bit), JDK 1.8, Eclipse 4.6.0
CPU	Intel core i7-4790, 3.60 GHz
Storage	8 G of memory, 1 TB disk

(1) MinR. It adopts the KM algorithm to assign tasks to candidate CSs that satisfy the user's privacy disclosure requirements, so as to minimize privacy risk without considering benefit. The MinR problem can be formulated as follows:

$$\text{Min} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} R[i,j] \times A[i,j] \tag{17}$$

subject to (13)–(16).

(2) MaxB. It adopts the KM algorithm to assign tasks to candidate CSs that satisfy the users' privacy disclosure requirements, so as to maximize benefit without considering privacy risk. The MaxB problem can be formulated as follows:

$$\text{Max} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} B[i,j] \times A[i,j] \tag{18}$$

subject to (13)–(16).

(3) Cplex. It solves the optimization problem proposed in Section 3.5 with IBM's CPLEX Optimizer v12.2 [49] and assigns tasks based on the solution.

5.1. Experimental Setting

To comprehensively evaluate IKM, we first compare its effectiveness with MinR and MaxB in experiment set #1, and then compare its efficiency with Cplex in experiment set #2. The detailed experimental settings of set #1 and set #2 are shown in Tables 5 and 6, respectively.

Table 5. The experimental setting of Set #1.

		m	*n*	*w*	*rt*	*npd*	*ub*
Set #1	Set #1.1	20, 40, . . . , 200	10, 20, . . . , 100	0.5	2	[1, 10]	2
	Set #1.2	40, 80, . . . , 400					
	Set #1.3	100	50	0.1, 0.2, . . . , 0.9	2	[1, 10]	2
	Set #1.4				4	[1, 10]	2
	Set #1.5				4	[11, 20]	2
	Set #1.6				2	[1, 10]	4

Table 6. The experimental setting of Set #2.

		m	*n*	*ub*
Set #2	Set #2.1	20, 40, . . . , 200	10, 20, . . . , 100	2
	Set #2.2			4
	Set #2.3	40, 80, . . . , 400		2
	Set #2.4			4

In set #1, we use various SSBM scenarios by changing six parameters in the experiments: (1) the number of tasks (n); (2) the number of CSPs (m); (3) the weight of the risk and benefit balance (w); (4) the privacy risk threshold (rt); (5) the number of private data required (npd), and (6) the service provision upper bound (ub). Specifically, in set #1.1, m changes from 20 to 200 with a step of 20, $n = m/2$, npd is randomly generated from the range [1, 10], w, rt, and ub are set to 0.5, 2, and 2, respectively. In set #1.2, m changes from 40 to 400 with a step of 40, and the other parameters are set as in set #1.1. In set # 1.3, n and m are fixed at 50 and 100 respectively, w changes from 0.1 to 0.9 with a step of 0.1, and the other parameters are set as in set # 1.1. In set # 1.4, rt is changed to 4, and the other parameters are set as in set # 1.3. In set #1.5, npd is randomly generated from the range [11,20], and the other parameters are set as in set # 1.4. In set #1.6, ub is expanded to 4, and the other parameters are set as in set #1.3.

Due to the solution time of the SSBM problem is mainly affected by the parameters n, m, and ub, in set #2, we conducted four sets of subexperiments by changing n, m, and ub. Specifically, in set #2.1, all parameter settings are the same as set #1.1. In set #2.2, ub is changed to 4, and other parameter settings are the same as set #2.1. In set #2.3, all parameter settings are the same as set #1.2. In set #2.4, ub is expanded to 4, and other parameter settings are the same as set #2.3.

In experiment sets #1 and #2, because not every CSP can offer CSs for all tasks, we generate the candidate CSs of each CSP randomly with 20–40% of n. The trust degree of each CSP is randomly generated in the range [0.25, 0.75]. The sensitivity degree of each private data value is randomly generated in the range [0.25, 0.75]. The privacy preferences and the privacy policies are randomly generated for each private data value. Specifically, for a privacy preference $pre_l = <pd_l, Pu_l, Lc_l, re_l>$, Pu_l is randomly generated from a purpose set containing 10 different purposes, Lc_l is randomly generated from a location set containing 10 different locations, and re_l is assigned randomly from 1–360 days. For a privacy policy $pol_{ik}^h = <pd_{ik}^h, Pu_{ik}^h, lc_{ik}^h, re_{ik}^h>$, Pu_{ik}^h, and re_{ik}^h are the same as the setting of corresponding privacy attributes in pre_l, and lc_{ik}^h is assigned randomly from 10 different locations. The parameters α and β of the benefit function are set to 0.5 and 10, respectively. In each experiment, 100 instances of experiments were run and the results are averaged.

5.2. Effectiveness Evaluation

To assess the effectiveness of IKM, we use four assessment indicators: the overall utility of a solution, the privacy risk of a solution, the benefit of a solution, and the benefit-to-risk ratio of a solution. For the ease of description, we use utility, risk, benefit, and benefit-to-risk ratio to represent the overall utility, privacy risk, benefit, and benefit-to-risk

ratio of a solution, respectively. Besides this, the risk described in the following experiments refers to the original privacy risk of a solution, i.e., the privacy risk before normalization.

Since MinR and MaxB only consider risk or benefit, respectively, the utility of MinR is actually a negative value for the normalized risk of the solution it finds, while the utility of MaxB is actually the benefit of the solution it finds. Furthermore, with the disclosure of private data, both MinR and MaxB obtain certain benefit and risk, and increase or decrease as the parameters n, m, w, npd, rt, and ub vary. In order to compare the margin of change in the risks and benefits of different approaches, we propose a benefit-to-risk ratio indicator, which is defined as the ratio of benefit to risk for a solution. The greater the benefit a solution obtains, the higher its benefit-to-risk ratio. Conversely, the higher the risk a solution takes, the lower its benefit-to-risk ratio.

Figures 6–11 present the effectiveness of IKM in experiment set #1 and the influence of six parameters, i.e., n, m, w, rt, npd, and ub. On the whole, IKM finds the optimal solution with a utility between MinR and MaxB. Furthermore, the risk and benefit of each approach vary with the changes of n, m, w, rt, npd, and ub. In most cases, the benefit-to-risk ratio of IKM is better than that of MinR and MaxB.

Figure 6 illustrates the effect of increasing m on the utility, risk, benefit, and benefit-to-risk ratio. As shown in Figure 6a, as m increases, the utilities of IKM and MaxB increase rapidly, while the utility of MinR gradually decreases. The reason is that n increases proportionally with the increase of m, resulting in more candidate CSs being selected. Furthermore, MinR always selects the candidate CSs with the lowest risk, which leads to low risk and low benefit, and its utility gradually decreases as m increases. MaxB always selects the candidate CSs with the highest benefit, which leads to high risk and high benefit, and its utility increases rapidly as m increases. IKM always selects the candidate CSs with the greatest utility by balancing the risk and benefit, resulting in its utility increaseing with the increase of m, and is between MinR and MaxB. For example, in Figure 6a, the average utilities of MinR, IKM, and MaxB are -1.29, 3.48, and 24.67, respectively.

In Figure 6b,c, the risks and benefits of all the approaches increase with the increase of m, and the growth rate of the benefit is faster than the growth rate of the risk. In Figure 6d, the benefit-to-risk ratios of all the approaches also increase with the increase of m. In all cases, MinR shows the lowest benefit-to-risk ratio, IKM shows the highest benefit-to-risk ratio, and MaxB's benefit-to-risk ratio is slightly lower than IKM. The reason behind this phenomenon is similar to Figure 6a. For example, in Figure 6d, the average benefit-to-risk ratios of MinR, IKM, and MaxB are 0.07, 0.30, and 0.27, respectively.

When m/n enlarges from 2 to 4, the result presented in Figure 7 is different from Figure 6. When Figure 7a is compared with Figure 6a, the average utilities of MinR, IKM, and MaxB significantly increase by 38.43%, 35.09%, and 7.25%, respectively. This is because as m enlarges to $4n$, the average range of candidate CSs for each task also enlarges. Similarly, if Figure 7b,c is compared with Figure 6b,c, the risks and benefits of IKM and MaxB increase at different margins, and, furthermore, the growth rate of benefits is faster than that of risks. In addition, the risk and benefit of MinR decrease at different margins, and the reduction rate of risk is faster than that of benefit. For example, if we compare Figure 7d with Figure 6d, we find that the average benefit-to-risk ratios of MinR, IKM, and MaxB increase by 18.78%, 10.26%, and 9.11%, respectively.

Figure 8 demonstrates the effect of w on the utility, risk, benefit, and benefit-to-risk ratio after fixing m and n. It can be seen from Figure 8a that, since MinR and MaxB only consider risk or benefit, respectively, their utilities are not affected by w, and are a fixed value in all cases, e.g., -1.24 and 21.86. However, the utility of IKM increases with the increase of w, and is always between MinR and MaxB. For example, in Figure 8a, the average utility of IKM is 5.38. Similarly, in Figure 8b,c, the benefit and risk of IKM also increase with the increase of w and are between MinR and MaxB. From Figure 8d, we can also see that the benefit-to-risk ratio of IKM is significantly higher than that of MinR. At the beginning, it increases rapidly with the increase of w, and reaches the maximum value when $w = 0.4$. Then it exceeds MaxB and decreases slightly as w continues to increase.

This is because when w is relatively small, the benefit growth rate of IKM is significantly faster than that of risk. When $w > 0.4$, the risk growth rate of IKM catches up with the benefit growth rate. In particular, when $w = 0.6$, the benefit of IKM has basically reached the upper limit, while its risk still continues to increase, resulting in a slight decrease in the benefit-to-risk ratio. It is worth noting that in all cases where $w > 0.4$, the benefit-to-risk ratio of IKM is still higher than that of MaxB.

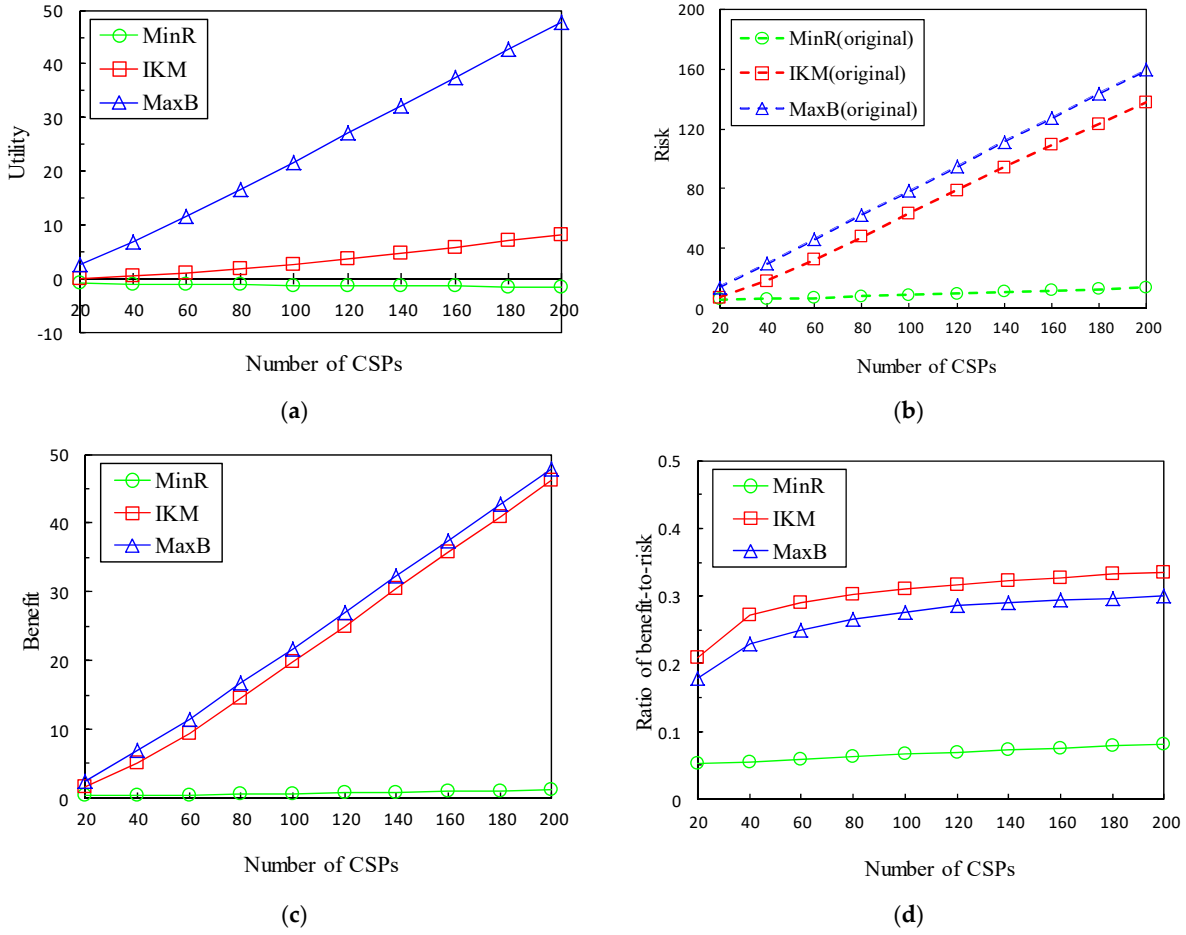


Figure 6. Effectiveness vs. number of CSPs (Set #1.1). (**a**) Utility; (**b**) risk; (**c**) benefit; (**d**) benefit−to−risk ratio.

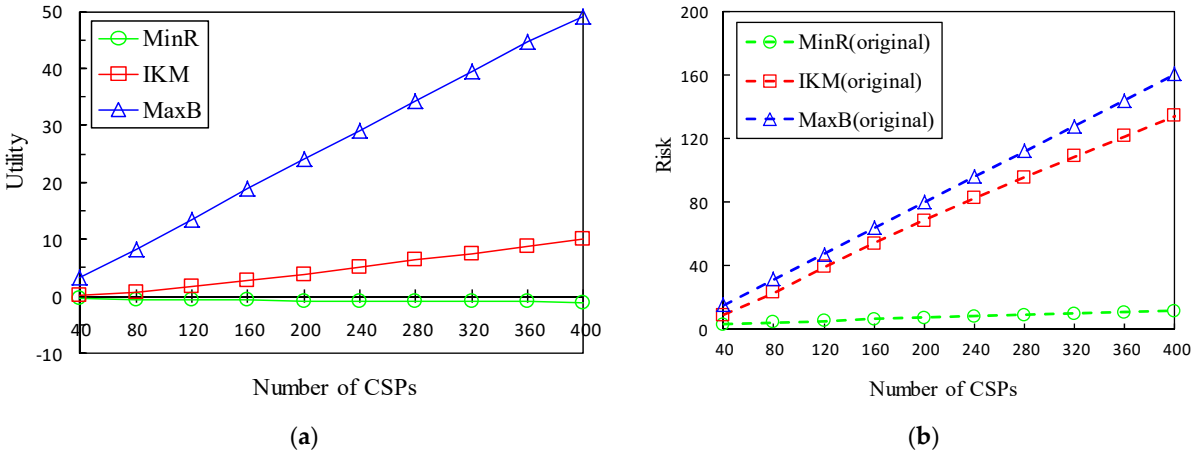
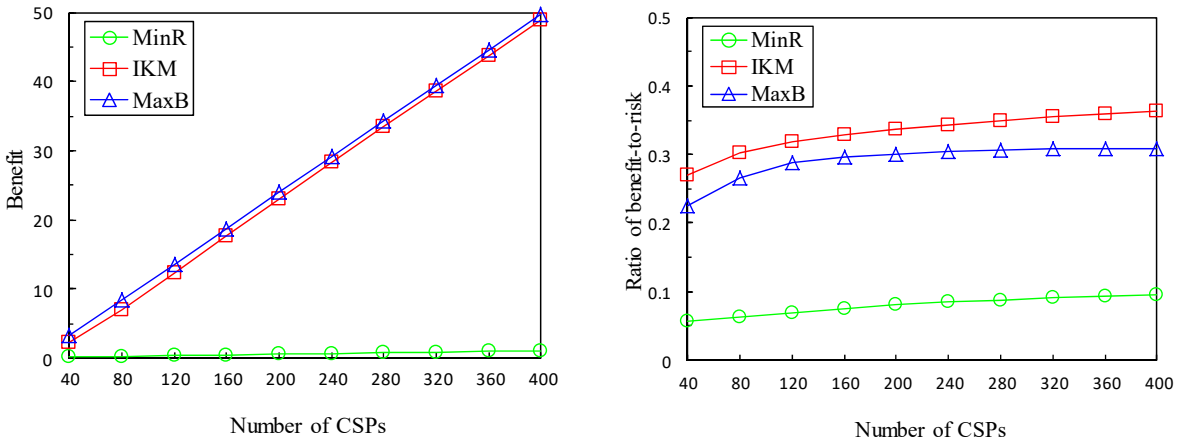
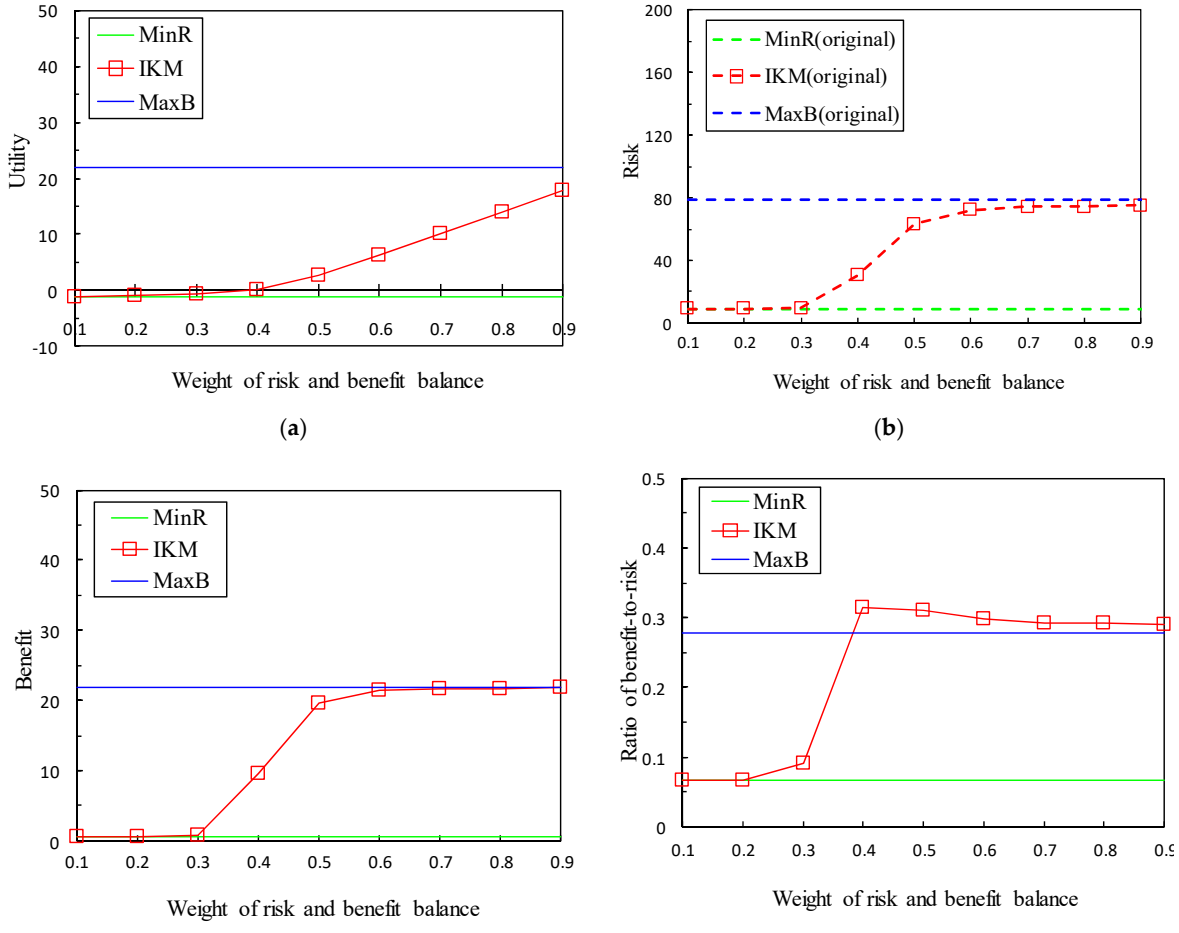


Figure 7. *Cont.*



(**c**) (**d**)

Figure 7. Effectiveness vs. number of CSPs (Set #1.2). (**a**) Utility; (**b**) risk; (**c**) benefit; (**d**) benefit−to−risk ratio.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Effectiveness vs. weight of risk and benefit balance (Set #1.3). (**a**) Utility; (**b**) risk; (**c**) benefit; (**d**) benefit−to−risk ratio.

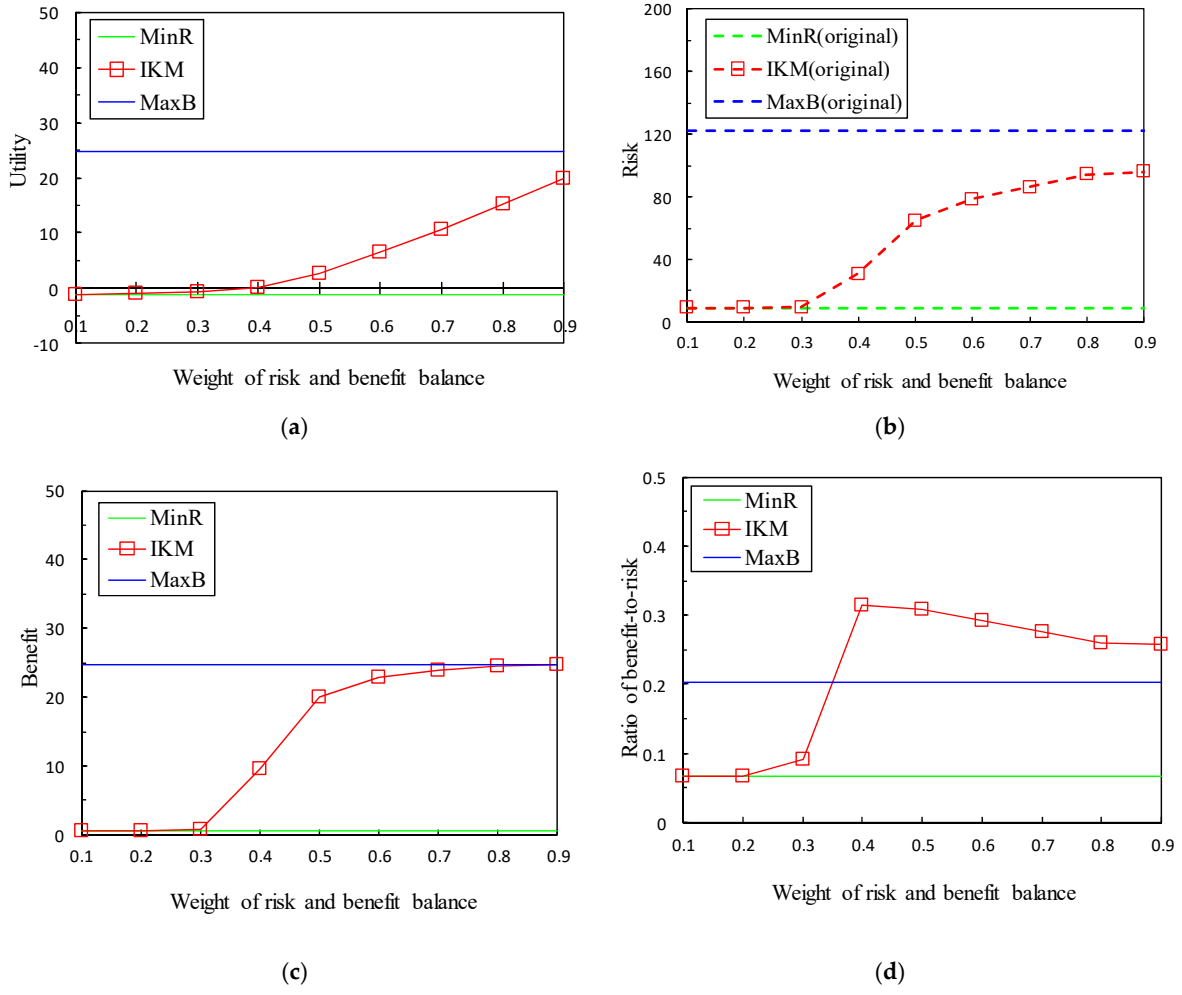


Figure 9. Effectiveness vs. weight of risk and benefit balance (Set #1.4). (**a**) Utility; (**b**) risk; (**c**) benefit; (**d**) benefit−to−risk ratio.

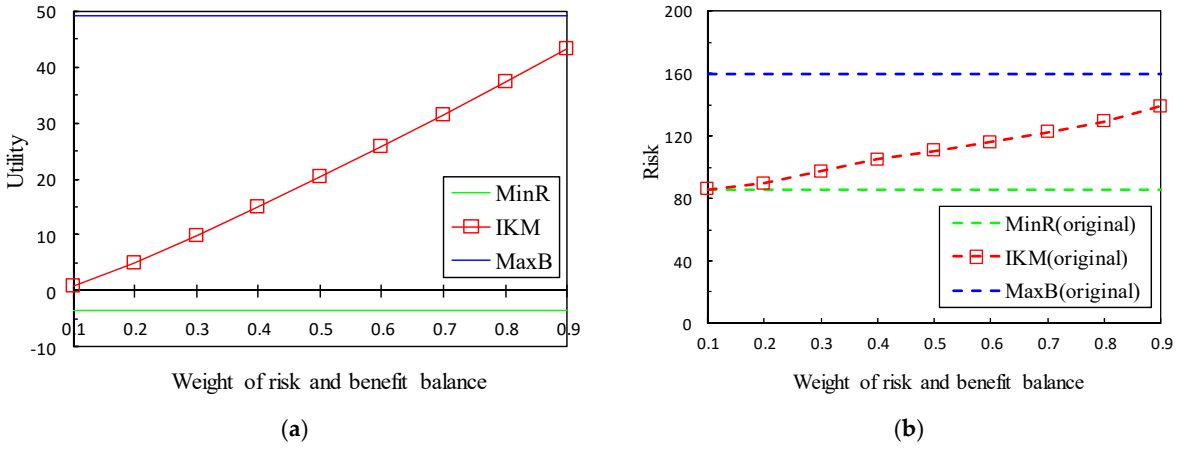
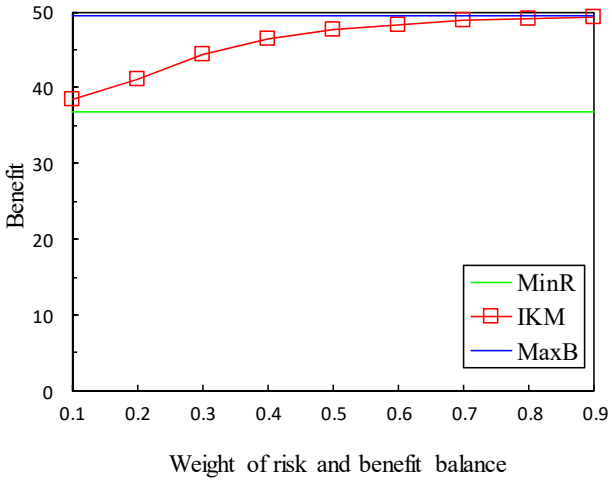
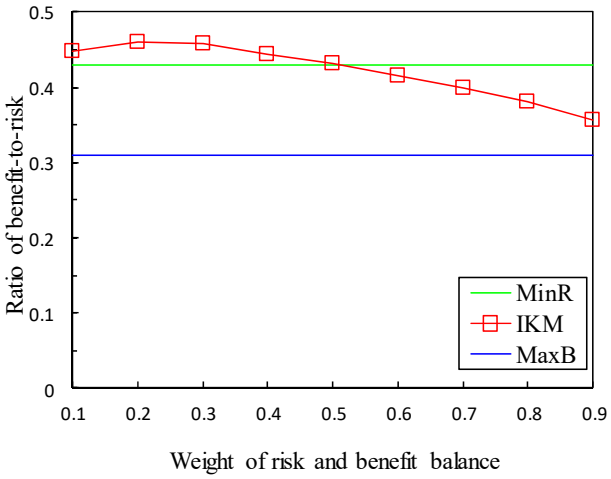


Figure 10. *Cont.*

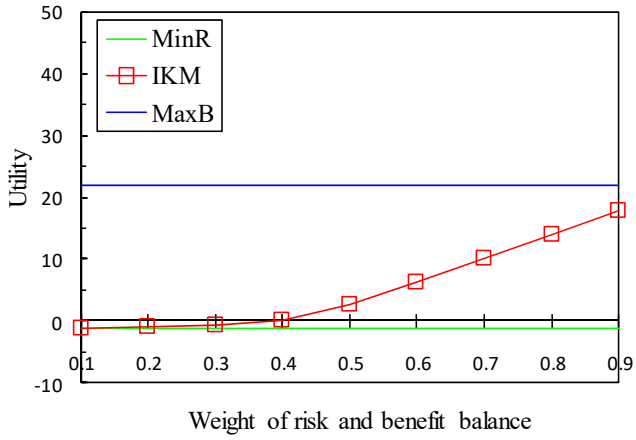


(**c**)

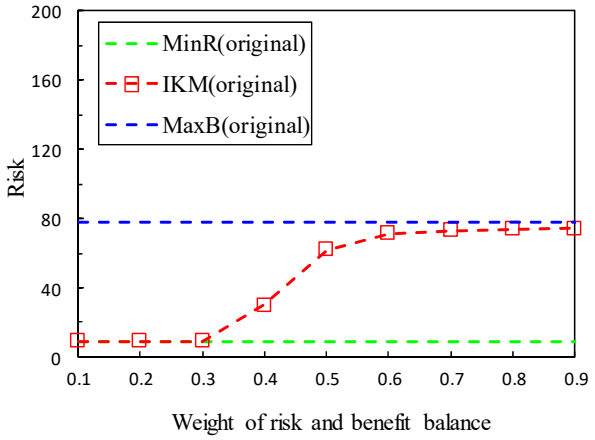


(**d**)

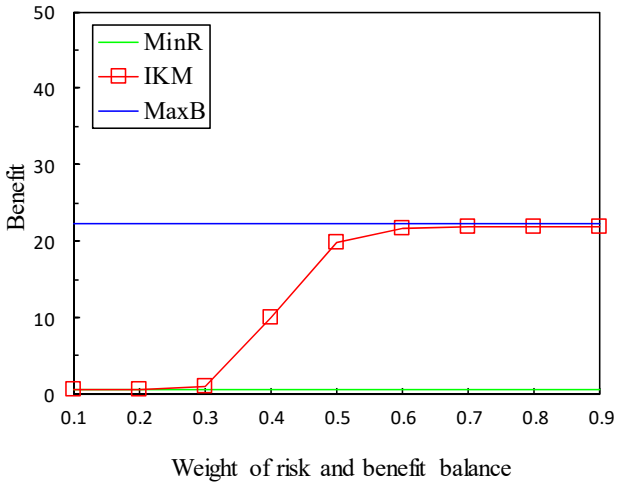
Figure 10. Effectiveness vs. weight of risk and benefit balance (Set #1.5). (**a**) Utility; (**b**) risk; (**c**) benefit; (**d**) benefit−to−risk ratio.



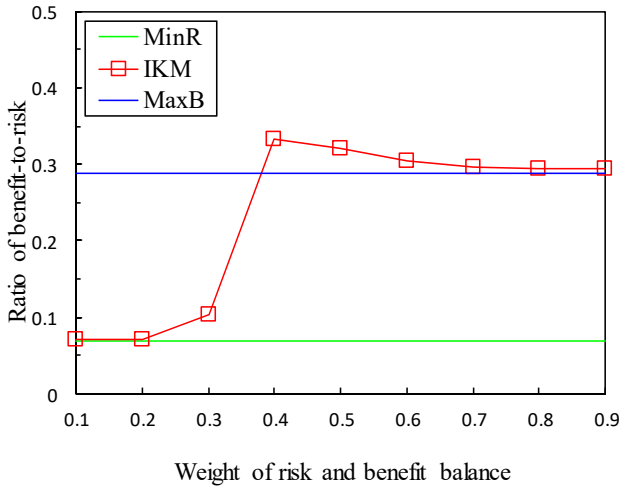
(**a**)



(**b**)



(**c**)



(**d**)

Figure 11. Effectiveness vs. weight of risk and benefit balance (Set #1.6). (**a**) Utility; (**b**) risk; (**c**) benefit; (**d**) benefit−to−risk ratio.

Figure 9 shows the effect of relaxing the *rt* constraint. Comparing Figure 9a with Figure 8a, the (average) utilities of MinR, IKM, and MaxB increase by 1.61%, 8.43%, and 13.04%, respectively. The reason is that the increase in *rt* relaxes the restrictions on the range of candidate CSs, which leads to an increase in the utility of all the approaches. Similarly, comparing Figure 9b,c with Figure 8b,c, the risks and benefits of all the approaches increase with the increase of *rt*, and the growth rate of risk is faster than that of benefit. In particular, the risk of MaxB increases significantly. It is because MaxB always selects the candidate CSs with the highest benefit, resulting in that more private data is disclosed, and the risk increases rapidly. When comparing Figure 9d with Figure 8d, the benefit-to-risk ratios of all the approaches decline to different degrees. Since the risk growth rate of MaxB is significantly faster than its benefit growth rate, in the case of $w \geq 0.4$, IKM further outperforms MaxB in terms of the benefit-to-risk ratio. For example, in Figure 9d, the (average) benefit-to-risk ratios of MinR, IKM and MaxB decrease by 0.6%, 6.76% and 27.09%, respectively.

Figure 10 illustrates the effect of raising the value range of *npd*. Comparing Figure 10a with Figure 8a, the (average) utilities of MaxB and IKM increase by 290.19% and 125.50%, respectively, while the utility of MinR decreases by 173.38%. This is because the increase in *npd* leads to an increase in the risks and benefits of all the approaches. At the same time, due to benefit being nonlinearly related to *npd*, especially when $npd > 10$, the increase of *npd* has a different effect on each approach. Specifically, compared with Figure 8b,c, in Figure 10b,c, the (average) risks of MinR, IKM and MaxB increase by 854.81%, 138.31%, and 102.35%, and the (average) benefits increase by 6020.66%, 250.36%, and 126.39%, respectively. It can be seen clearly that the growth rate of risk is significantly faster than that of benefit. Therefore, comparing with Figure 8d, the (average) risk–benefit-ratios of MinR, IKM and MaxB in Figure 10d increase by 541.72%, 87.32%, and 11.90%, respectively. It is worth noting that in all cases, the benefit-to-risk ratio of IKM is significantly higher than MaxB, and in the case of $w \leq 0.5$, its benefit-to-risk ratio also exceeds MinB. However, as w continues to increase, its benefit-to-risk ratio gradually declines and is lower than MinB. This is because when $w \leq 0.5$, the benefit growth rate of IKM is significantly faster than its risk growth rate, and when $w > 0.5$, the benefit of IKM has basically reached its upper limit, but its risk still grows rapidly, resulting in a decline in the benefit-to-risk ratio.

Figure 11 depicts the effect of increasing *ub*. When *ub* increases from 2 to 4, we can compare Figure 11a with Figure 8a. The (average) utilities of MinR, IKM, and MaxB all slightly increase, i.e., 3.22%, 3.65%, and 3.43%. Similarly, when comparing with Figure 8d, in Figure 11d, the (average) benefit-to-risk ratios of all the approaches also increase slightly, i.e., 3.12%, 3.38%, and 3.86%. The reason behind these results is that increasing *ub* expands the range of candidate CSs, thus improving the utility and benefit-to-risk ratio. We need to point out that when $ub = 2$, the capacity of candidate CSs meets the need to obtain the optimal solution. Thus, increasing *ub* does not significantly affect the utility, risk, benefit, and benefit-to-risk ratio.

5.3. Efficiency Evaluation

To evaluate the efficiency of IKM, we compared the average execution time of IKM and Cplex in solving the SSBM problems. Figures 12 and 13 present the time taken by all the approaches to find a solution and the effects of n, m, and *ub* in experiment set #2. Generally, and overall, IKM is much faster than Cplex, and the larger the m, the more obvious this trend.

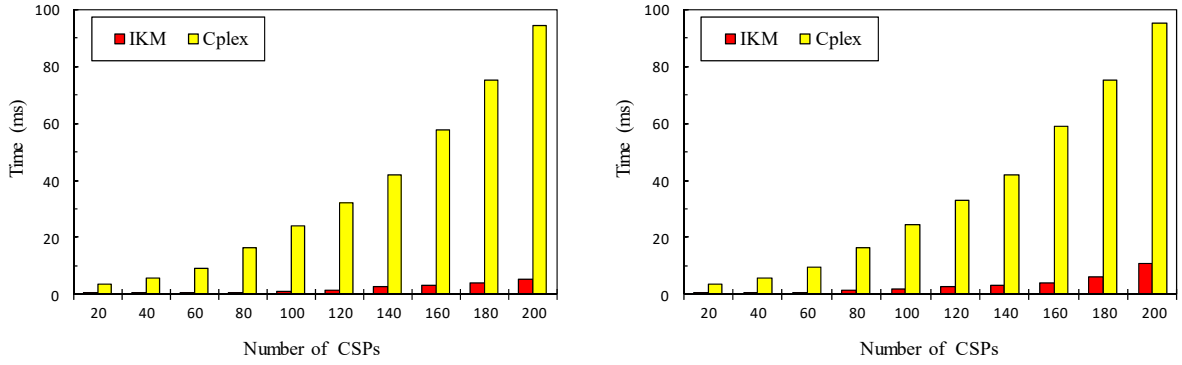


Figure 12. Average time consumption vs. number of CSPs (Sets #2.1 and #2.2). (**a**) Set #2.1; (**b**) Set #2.2.

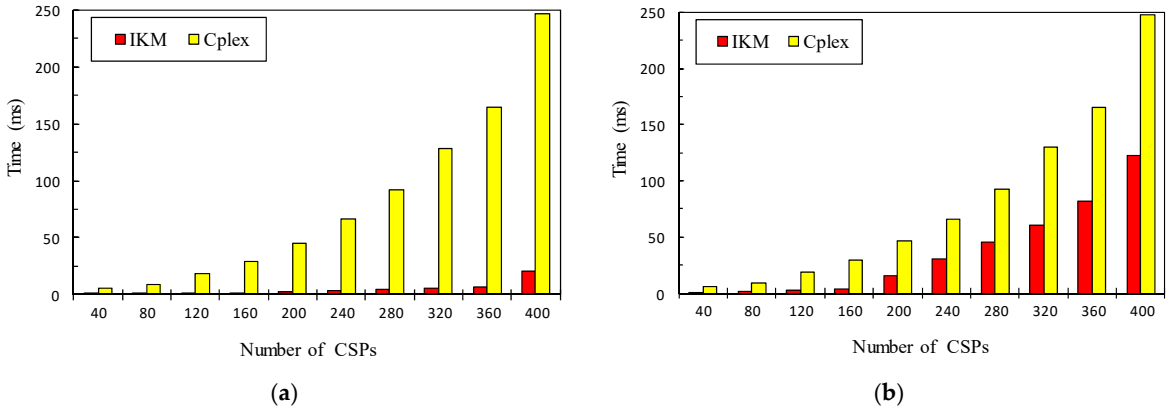


Figure 13. Average time consumption vs. number of CSPs (sets #2.3 and #2.4). (**a**) Set #2.3; (**b**) set #2.4.

As shown in Figure 12a,b, a larger m consumes more time, but Cplex shows a much more rapid growth trend than IKM in all cases. In Figure 12a, when m increases from 20 to 200, the time taken by Cplex increases from 3.39 ms to 94.33 ms, while the consumed time of IKM only increases from 0.15 ms to 5.11 ms. The results observed from Figure 12b show the influence of increasing ub on time consumption. If we compare Figure 12b with Figure 12a, we notice that ub expands from 2 to 4 and the execution time of IKM shows a rapid growth trend, especially in some cases where m is relatively large. However, the execution time of Cplex remains basically the same and is still much longer than that of IKM. For example, in Figure 12b, when m increases from 20 to 200, the time taken by Cplex increases from 3.57 ms to 95.24 ms, while the consumed time of IKM only increases from 0.2 ms to 10.89 ms. The reason is that the search space of IKM expands with the enlargement of ub. A lucky situation is that ub is usually very small, e.g., $ub = 2$.

Figure 13 shows the efficiency results in experiment sets # 2.3 and 2.4. In this set of experiments, as m enlarges to $4n$, the time consumption of all the approaches further increases compared to experimental sets # 2.1 and 2.2, but the growth trend of Cplex is faster. Comparing Figure 13a with Figure 12a, the time consumption significantly increases. For example, when m increases from 40 to 400, the consumed time of Cplex increases from 5.33 ms to 246.22 ms, while the time taken by IKM only increases from 0.22 ms to 20.33 ms. As ub expands from 2 to 4, Figure 13b replaces Figure 12b, the consumed time of all the approaches further increases. They are similar to those of Figure 12b, e.g., when m increases from 40 to 400, the consumed time of Cplex increases from 6.02 ms to 247.49 ms, while the time consumed by IKM increases from 0.3 ms to 122.59 ms.

5.4. Discussion

From the above experimental results, we can make the following conclusions.

(1) IKM, MinR, and MaxB take the same execution time to solve their problems, and the utility of IKM is between MinR and MaxB. However, in most cases, IKM is better than MinR and MaxB when comparing the actual benefit received with the risk assumed. Additionally, IKM and Cplex have the same result in solving the SSMB problem. However, in terms of performance, the time overhead of IKM is much smaller than that of Cplex. Thus, comprehensively comparing these approaches, IKM is considered to be a better approach to finding the optimal solution to the SSBM problem.

(2) As *npd* increases, the utilities of IKM and MaxB increase, while the utility of MinR decreases. Therefore, for users who balance privacy risk and benefit and users who only care about benefit, they can obtain more benefits by appropriately selecting CSs that require more private data, while for users who only care about privacy, they should select as many CSs as possible that require less private data.

(3) When *npd* is relatively small, the benefit-to-risk ratio of IKM increases as w increases. It is higher than MinR in all cases, and exceeds MaxB when $w \geq 0.4$. However, in the case of relatively large *npd*, the benefit-to-risk ratio of IKM is always higher than MaxB, it decreases with the increase of w, and is lower than MinR when $w > 0.5$. Therefore, for users who balance privacy risk and benefit, they should adjust w according to a different *npd* to obtain a higher benefit-to-risk ratio.

(4) Although expanding m/n and *ub* can improve the utility and benefit-to-risk ratio, it also brings more time consumption. In addition, relaxing the *rt* constraint can increase utility, but it also leads to a decrease in the benefit-to-risk ratio. In summary, in service selection, privacy disclosure requirements can be set by a user according to the utility, profit-to-risk ratio, and time consumption with alternative combinations of m, n, w, *npd*, *rt*, and *ub*.

(5) Since expanding m/n can increase the utility and benefit-to-risk ratio of all the approaches, CSPs should increase the supply of candidate service types in order to obtain more service provision opportunities. Additionally, because different user groups have different privacy disclosure requirements, CSPs should also disclose private data and provide personalized services based on the user groups that provide services.

(6) For any two tasks in MPCSC, if they are assigned to services provided by the same CSP, then the CSP will collect multiple pieces of private data from the user, and can infer more privacy information from the collected data. Therefore, this type of task assignment will increase the overall privacy risk of a CSC. In order to calculate this part of the increased privacy risk, it needs to expand the objective function of the SSBM problem. The expanded SSBM problem is as follows:

$$\text{Max} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} U[i,j] \times A[i,j] + \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \sum_{j'=0}^{n-1} R*[i,j,j'] \times A[i,j] \times A[i,j'] \quad (19)$$

subject to (13)–(16).

where R^* is an $m \times n \times n$ matrix, $R^*[i, j, j'] \in [0, 1]$ express the increasing degree of task j's risk when task j is assigned to CSP i and task j' is also assigned to CSP i, $i \in \{0, 1, \ldots, m-1\}$, $j \in \{0, 1, \ldots, n-1\}$, $j' \in \{0, 1, \ldots, n-1\}$. The extended SSMB problem is a nonlinear 0–1 programming problem, it has a higher complexity than the SSMB problem and cannot be solved by the IKM approach. Cplex can be used to solve this problem [50]. We did not consider this problem in the current paper and plan to address it in future work.

(7) The service selection approach proposed in this paper is practical and feasible. As shown in Figure 1, users only need to submit their functional requirements and privacy requirements to the cloud service broker. The cloud service broker can utilize the cloud service composition manager to discover candidate CSs from different CSPs. The cloud service composition manager then evaluates the privacy risks and personalized benefits of these candidate CSs, and selects a set of CSs with the maximum utility for users to use. The abovementioned service discovery, evaluation, and selection processes are automatically completed by the cloud service composition manager, and do not require user participation. In future work, we intend to develop a web

application-based service selection tool for the proposed approach, where users can set their functionality and privacy requirements by simply filling in and selecting some parameter values in web pages. Therefore, the users can use it very easily.

(8) In MPCSC, each CSP can provide services for multiple tasks and expect users to select as many services as possible. However, if a user selects too many services from the same CSP, it will lead to serious privacy leakage risks, and will also face high service prices and vendor lock-in. The proposed approach selects CSs from multiple CSPs, which can effectively reduce privacy risks and obtain services with lower prices and higher quality. At the same time, the proposed approach can also motivate CSPs to continuously enhance privacy protection capabilities, reduce asking prices, and improve service quality to compete for more service provision opportunities. Indeed, selecting services from multiple CSPs may incur some financial cost as well as technical difficulties compared to selecting services from a single CSP. However, in general, the benefits of selecting services from multiple CSPs can offset these costs, and furthermore, as can be seen from our simulation experiments, the proposed approach is technically practical and feasible.

6. Related Work

6.1. Service Selection for MPCSC

With the emergence of a large number of CSPs that can offer multiple services on the web, the service selection for MPCSC has become more complicated. Recently, many research efforts have been focusing on service selection based on the relationship between services, such as bundled service provision, complementarity between services and QoS correlation between services. In [2], He et al. proposed a composition service selection framework, where an iterative multiattribute combined auction model is employed to select services, and the complementarity between services is considered. Aiming at the QoS correlation between services provided by the same service provider, Deng et al. [24] put forward a correlation-aware service pruning method in the selection of candidate services. Targeting the hybrid quality correlation between services provided by different service providers, Zhang et al. [25] proposed a quality correlation query approach for service composition, which uses a quality correlation index graph to achieve efficient queries for quality correlations.

Due to the many advantages of using services and resources from multiple cloud providers, e.g., access to distributed resources, avoiding vendor lock-in, and high application elasticity, more and more cloud customers are migrating their applications from a single cloud provider to multiple cloud providers. Some researchers have started to explore service selection and deployment issues in a hybrid cloud, federated cloud, and multicloud. Ma et al. [7] proposed a CS composition approach for data-intensive applications in a hybrid cloud. This approach formalizes the CS selection problem based on the conflict and collaboration relationship between CSs, and uses the IBM ILOG CPLEX optimization solution to solve the problem. Considering the requirements of reliability, security, cost of computing power, data storage, and inter-cloud communication, Wen et al. [26] presented an approach for deployment workflow applications on federated clouds. To deal with the deployment problem of multiple composite applications, Shi et al. [8] put forward a composite applications deployment approach based on the GA algorithm, which selects appropriate CSs from multiple clouds for multiple composite applications under cost budget constraint, so that the average response time is minimized.

The above work considers the service selection problem in MPCSC based on the relationship between services and different service provision strategies. These works select the optimal candidate services from the perspective of satisfying QoS and resource constraints, while ignoring the privacy protection requirements. However, in MPCSC, privacy is a critical user requirement.

6.2. Privacy-Preserving Service Selection

In recent years, users have increasingly paid attention to the privacy of their personal data. How to select the best services for service composition and meet the user privacy requirements has become a critical issue. Several research contributions have focused on service selection and composition that protects user privacy. These works usually use automated technology to match the provider's privacy policy with the user's privacy preferences, and then select a set of services with the highest matching degree to build service composition. The work in [27] presented a privacy-aware service composition and ranking framework, which verifies the compliance between users' privacy requirements and services' privacy policies, and selects the service composition with the highest privacy level from the verified multiple service composition schemes. Meng et al. [31] put forward a privacy-aware cloud service selection approach that models the users' preferences and CSs' privacy policies, and recommended a set of privacy trusted CSs to users by a policy-matching algorithm. Bharati et al. [29] proposed a method to integrate Blockchain technique and GDPR rules to enhance cloud user privacy, which records a set of operations performed by CSPs on data in the Blockchain network, and verifies whether the operations performed by CSPs comply with GDPR requirements. Targeting the service selection problem in the data as a service (DaaS) composition, Tbahriti et al. [30] presented a privacy model that specifies the privacy requirements and policies of the services, and verified the compatibility between privacy requirements and policies in DaaS composition. Aiming at the access control requirements and privacy requirements in CSC, Amini et al. [28] combined an attribute-based access control model and a purpose-based privacy model to protect the privacy of users while controlling access to services by unknown users.

To help the users select privacy-sensitive IoT services in a smart environment, Alom et al. [32] formulated the privacy checking problem as a knapsack problem, and proposed both knapsack privacy checking techniques and knapsack graph-based privacy checking techniques to solve the problem, thereby recommending a suitable set of IoT services to users. Targeting the privacy-preserving workflow scheduling problem in geographically distributed data centers, Xiao et al. [51] proposed a privacy-preserving workflow scheduling algorithm, which aimed to minimize the data transfer time between data centers for workflows while satisfying multilevel data privacy requirements.

To deal with the service selection issue based on privacy risk, Yu et al. [34] defined the privacy risk aware service selection problem as a multiconstrained optimal path problem, and used the extended MCSP/MCSP-K algorithms to find an optimal solution that satisfies the QoS and privacy constraints. Similarly, Belabed et al. [35] formulated the privacy protection web service selection problem as a multiple-choice knapsack problem, and proposed three methods, i.e., based on the best first search algorithm, based on proposition satisfiability, and based on answer set programming, to solve the composite service selection problem with the least privacy risk.

In our previous work [6], an approach for privacy regulation-aware CS selection for MPCSC is presented, and the CS selection problem is modeled as an optimization problem with privacy constraints, where an optimal solution can be found by pre-processed KM algorithm.

Although the above work has merit, the privacy protection composite service selection is still an open issue. Table 7 compares some related work with our proposed method, including the privacy requirements, privacy policy matching, privacy risk measurement, personalized benefit measurement, and multiservice provision characteristics. In Table 7, we find that the mentioned work can express part of the privacy requirements, and support the matching checks between privacy policies and privacy requirements, but their privacy requirements rarely fully consider privacy attributes such as sensitivity, purpose, location, and retention. Moreover, several works have focused on privacy risk measurement for services, but they generally do not consider the potential benefit of disclosing private data. Furthermore, few research efforts paid attention to the multiservice provision characteristic of the CSPs. Different from existing methods, our proposed method can express the users'

privacy requirements more comprehensively, support quantitative measures of privacy risk and personalized benefit, and also consider the multiprovision characteristic of the CSPs. Therefore, the proposed method is able to trade-off privacy risk and personalized benefitin service selection according to the user's preferences.

Table 7. The comparison of privacy-preserving service selection methods.

Works	Privacy Requirements				Privacy Policy Matching	Privacy Risk	Personalized Benefit	Multiservice Provision
	Sensitivity	Purpose	Location	Retention				
Costante [27]	Yes	Yes	No	Yes	Yes	Yes	No	No
Meng [31]	Yes	Yes	No	No	Yes	No	No	No
Tbahriti [30]	No	Yes	No	Yes	Yes	No	No	No
Amini [28]	Yes	Yes	No	Yes	Yes	No	No	Yes
Barati [29]	No	Yes	No	No	Yes	No	No	No
Alom [32]	No	Yes	No	Yes	Yes	No	No	No
Yu [34]	No	No	No	No	No	Yes	No	No
Belabed [35]	No	Yes	No	Yes	Yes	Yes	No	No
Xiao [51]	Yes	No	No	No	No	No	No	Yes
Liu [6]	Yes	Yes	Yes	Yes	Yes	No	No	Yes
This work	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

6.3. Risk–Benefit Balance in Private Data Disclosure

Focusing on the privacy risks and social benefits trade-off of information sharing in online social networks, Yang et al. [52] put forward a utility-based trade-off framework that models the privacy risks and social benefits of data sharing, and maximizes the overall sharing utility of users by balancing privacy risks and benefits. Similarly, Sourya et al. [53] proposed an integer-programming model to help users make private data disclosure decisions in social networks. This model provides users with privacy settings suggestions for their profile attributes, so that users can obtain the greatest social benefits while avoiding some privacy risks.

To address the problem of data sharing that balances privacy risks and benefits in internet-of-things (IoT) applications, Mahmoud et al. [54] proposed a user-centric privacy-protection IoT data sharing architecture, which provides a data-sharing model that balances privacy risk and potential benefit to help users determine the degree of data sharing.

Focusing on the cost and benefit of private data disclosure in e-commerce transactions, Zhu et al. [11] proposed a cost–benefit analysis approach for private data disclosure based on the multiattribute utility theory (MAUT). This approach quantitatively evaluates the disclosure costs and benefits of private data, and recommends the greatest utility e-commerce company to users for transactions by balancing the costs and benefit.

Aiming at the privacy risks and benefits of personal data release in web services, Bikash et al. [55] presented a personal data storage architecture that enables users to store their private data in a unified data repository. Based on this architecture, the authors designed a set of access control policies based on the trade-off of privacy risks and benefits, and controlled the release of data through these policies.

The above work has conducted research on the issue of private data disclosure that balances risk and benefit in social networks, IoT, e-commerce, and web services. Compared with the above work, our work not only considers the risk–benefit balance in private data disclosure, but also considers the service selection in MPCSC with privacy constraints.

Role-based collaboration (RBC) has been proposed as a promising complex problem-solving methodology, and its environments—classes, agents, roles, groups, and objects (E-CARGO) model has been verified as a powerful model for complex systems. Zhu et al. [48] formalize the group role assignment (GRA) problem and propose an efficient solution by adapting the KM algorithm. The RBC process is a generalized problem-solving process, which can be used to illustrate many industry problems. GRA as a general problem model can provide templates for modeling many application problems. RBC and E-CARGO provide us with good guidelines and inspirations to investigate the SSBM problem discussed in this paper.

7. Conclusions

The users enjoy various benefits provided by personalized services, but at the cost of disclosing more private data. Especially in MPCSC, when a CSP provides multiple services at the same time, it may infer the previously hidden private information through the collected multiple private data, which may lead to more serious privacy leaks. Therefore, how to select a set of best candidate CSs with tradeoffs between privacy risk and benefit, and meanwhile meet the user's privacy disclosure requirements, remains challenging for the design of the MPCSC.

In this work, we present the privacy disclosure requirements in MPCSC according to the privacy preferences of users and the multiservice provision characteristic of CSPs. In consideration of the requirement of privacy disclosure, we quantitatively assessed the privacy risks and potential benefits of CSs, and formalized the SSBM problem as an integer programming optimization problem. Because the KM algorithm cannot be used to solve the SSBM problem directly, we designed an improved KM algorithm, and compared it with three benchmark approaches, i.e., MinR, MaxB and Cplex. Experimental results show:

(1) IKM, MinR, and MaxB have the same time performance. But in terms of utility, IKM is between MinR and MaxB, and in terms of benefit-to-risk ratio, IKM outperforms MinR and MaxB in most cases.
(2) IKM and Cplex have the same result in solving the SSMB problem. However, in terms of performance, the time consumption of IKM is much less than that of Cplex.

Hence, comprehensively comparing these approaches, IKM is considered as a better approach to finding the optimal solution to the SSBM problem.

For future work, we will make the service selection process more adaptable, so that if the user's privacy disclosure requirements and the cloud service environment change over time, the service selection process will automatically adjust to achieve the new optimal system utility. At the same time, the process of dynamically selecting services from different CSPs according to user privacy requirements is very complicated and requires certain costs, such as time cost, performance cost, etc. Therefore, another future direction is to evaluate the cost-of-service selection, and use it as a basis for the user to decide whether to start a new service selection process. Moreover, since this paper mainly analyzes the service selection process through simulation experiments, we plan to further automate and transparent this process in future work to provide a service selection tool that can be used in real-world scenarios. Furthermore, we also plan to apply the proposed approach to other related applications, such as crowdsourcing task assignment, big data task placement, service selection in edge computing and IoT, etc.

Author Contributions: Conceptualization, L.L. and H.Z.; methodology, L.L. and H.Z.; software, L.L. and S.C.; validation, L.L. and S.C.; formal analysis, L.L. and H.Z.; investigation, L.L. and S.C.; resources, H.Z.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, L.L. and H.Z.; visualization, S.C.; supervision, H.Z.; project administration, L.L.; funding acquisition, L.L. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Sciences and Engineering Research Council of Canada (Grant No. RGPIN-2018-04818), and Jiangsu Province Planning Subject for the 13th Five Year Plan of Education Sciences (Grant No. 2016-GH0303-00022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors would like to thank the editor and the anonymous reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.H.; Andrew, K.; Gunho, L.; David, A.P.; Ariel, R.; Stoica, I.; et al. Above the Clouds: A Berkeley View of Cloud Computing. *Comm. ACM* **2010**, *53*, 50–58. [CrossRef]
2. He, Q.; Yan, J.; Jin, H.; Yang, Y. Quality-Aware Service Selection for Service-Based Systems Based on Iterative Multi-Attribute Combinatorial Auction. *IEEE Trans. Softw. Eng.* **2014**, *40*, 192–215. [CrossRef]
3. Jamshidi, P.; Ahmad, A.; Pahl, C. Cloud Migration Research: A Systematic Review. *IEEE Trans. Cloud Comput.* **2013**, *1*, 142–157. [CrossRef]
4. Gartner. Gartner Forecasts Worldwide Public Cloud Revenue to Grow 6.3% in 2020. 2020. Available online: https://www.gartner.com/en/newsroom/press-releases/2020-07-23-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-6point3-percent-in-2020 (accessed on 10 April 2022).
5. Lin, D.; Squicciarini, A.C.; Dondapati, V.N.; Sundareswaran, S. A Cloud Brokerage Architecture for Efficient Cloud Service Selection. *IEEE Trans. Cloud Comput.* **2019**, *12*, 144–157. [CrossRef]
6. Liu, L.; Zhu, H.; Chen, S.; Huang, Z. Privacy Regulation Aware Service Selection for Multi-provision Cloud Service Composition. *Future Gener. Comput. Syst.* **2022**, *126*, 263–278. [CrossRef]
7. Ma, H.; Zhu, H.; Li, K.; Tang, W. Collaborative Optimization of Service Composition for Data-Intensive Applications in a Hybrid Cloud. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *30*, 1022–1035. [CrossRef]
8. Shi, T.; Ma, H.; Chen, G.; Sven, H. Location-Aware and Budget-Constrained Service Deployment for Composite Applications in Multi-Cloud Environment. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1954–1969. [CrossRef]
9. Pallant, J.I.; Pallant, J.L.; Sands, S.J.; Ferraro, C.R.; Afifi, E. When and How Consumers Are Willing to Exchange Data with Retailers: An Exploratory Segmentation. *J. Retail. Consum. Serv.* **2022**, *64*, 1–12. [CrossRef]
10. Gerber, N.; Gerber, P.; Volkamer, M. Explaining the Privacy Paradox: A Systematic Review of Literature Investigating Privacy Attitude and Behavior. *Comput. Secur.* **2018**, *77*, 226–261. [CrossRef]
11. Zhu, H.; Ou, C.X.; Van den Heuvel, W.J.A.M.; Liu, H. Privacy Calculus and Its Utility for Personalization Services in E-commerce: An Analysis of Consumer Decision-Making. *Inf. Manag.* **2017**, *54*, 427–437. [CrossRef]
12. Ghorbel, A.; Ghorbel, M.; Jmaiel, M. Privacy in Cloud Computing Environments: A Survey and Research Challenges. *J. Supercomput.* **2017**, *73*, 2763–2800. [CrossRef]
13. Bahri, L.; Carminati, B.; Ferrari, E. Privacy in Web Service Transactions: A Tale of More than a Decade of Work. *IEEE Trans. Serv. Comput.* **2018**, *11*, 448–465. [CrossRef]
14. Koh, B.; Raghunathan, S.; Nault, B.R. An Empirical Examination of Voluntary Profiling: Privacy and Quid Pro Quo. *Decis. Support Syst.* **2020**, *132*, 1–11. [CrossRef]
15. Awad, N.F.; Krishnan, M.S. The Personalization Privacy Paradox: An Empirical Evaluation of Information Transparency and The Willingness to be Profiled Online for Personalization. *MIS Q.* **2006**, *30*, 13–28. [CrossRef]
16. Dinev, T.; Hart, P. An Extended Privacy Calculus Model for E-commerce Transactions. *Inf. Syst. Res.* **2006**, *17*, 61–80. [CrossRef]
17. Ozturk, A.B.; Nusair, K.; Okumus, F.; Singh, D. Understanding Mobile Hotel Booking Loyalty: An Integration of Privacy Calculus Theory and Trust-Risk Framework. *Inf. Syst. Front.* **2017**, *19*, 753–767. [CrossRef]
18. Kordzadeh, N.; Warren, J. Communicating Personal Health Information in Virtual Health Communities: An Integration of Privacy Calculus Model and Affective Commitment. *J. Assoc. Inf. Syst.* **2017**, *18*, 45–81. [CrossRef]
19. Wahab, O.A.; Bentahar, J.; Otrok, H. Towards trustworthy multi-cloud services communities: A trust-based hedonic coalitional game. *IEEE Trans. Serv. Comput.* **2018**, *11*, 184–201. [CrossRef]
20. Nesrine, K.; Maryline, L.; Sana, B. Privacy Enhancing Technologies for Solving the Privacy-Personalization Paradox: Taxonomy and Survey. *J. Netw. Comput. Appl.* **2020**, *171*, 1–32.
21. Badsha, S.; Yi, X.; Khalil, I.; Liu, D.; Nepal, S.; Bertino, E.; Lam, K.Y. Privacy Preserving Location-Aware Personalized Web Service Recommendations. *IEEE Trans. Serv. Comput.* **2021**, *14*, 791–804. [CrossRef]
22. Kosinski, M.; Stillwell, D.; Graepel, T. Private Traits and Attributes are Predictable from Digital Records of Human Behavior. *Proc. Natl. Acad. Sci. USA* **2013**, *110*, 5802–5805. [CrossRef] [PubMed]
23. Cai, Z.; He, Z.; Guan, X.; Li, Y. Collective Data-Sanitization for Preventing Sensitive Information Inference Attacks in Social Networks. *IEEE Trans. Depend. Secur. Comput.* **2018**, *15*, 577–590. [CrossRef]
24. Deng, S.; Wu, H.; Hu, D.; Zhao, J.L. Service Selection for Composition with QoS Correlations. *IEEE Trans. Serv. Comput.* **2016**, *9*, 291–303. [CrossRef]
25. Zhang, Y.; Cui, G.; Deng, S.; Chen, F.; Wang, Y.; He, Q. Efficient Query of Quality Correlation for Service Composition. *IEEE Trans. Serv. Comput.* **2021**, *14*, 695–709. [CrossRef]
26. Wen, Z.; Jacek, C.; Paul, W.; Alexander, B.R. Cost Effective, Reliable and Secure Workflow Deployment over Federated Clouds. *IEEE Trans. Serv. Comput.* **2017**, *10*, 929–941. [CrossRef]
27. Costante, E.; Paci, F.; Zannone, N. Privacy-Aware Web Service Composition and Ranking. In Proceedings of the 2013 IEEE International Conference on Web Services, Santa Clara, CA, USA, 28 June–3 July 2013; pp. 131–138.
28. Amini, M.; Osanloo, F. Purpose-Based Privacy Preserving Access Control for Secure Service Provision and Composition. *IEEE Trans. Serv. Comput.* **2019**, *12*, 604–620. [CrossRef]
29. Barati, M.; Rana, O. Tracking GDPR Compliance in Cloud-based Service Delivery. *IEEE Trans. Services Comput.* **2018**. *to be published*. Available online: https://ieeexplore.ieee.org/document/9106853 (accessed on 10 April 2022). [CrossRef]

30. Tbahriti, S.E.; Ghedira, C.; Medjahed, B.; Mrissa, M. Privacy-Enhanced Web Service Composition. *IEEE Trans. Serv. Comput.* **2014**, *7*, 210–222. [CrossRef]
31. Meng, Y.; Huang, Z.; Zhou, Y.; Ke, C. Privacy-Aware Cloud Service Selection Approach Based on P-Spec Policy Models and Privacy Sensitivities. *Future Gener. Comput. Syst.* **2018**, *86*, 1–11. [CrossRef]
32. Alom, M.Z.; Singh, B.C.; Aung, Z.; Azim, M.A. Knapsack Graph-Based Privacy Checking for Smart Environments. *Comput. Secur.* **2021**, *105*, 1–15. [CrossRef]
33. Union, E. General Data Protection Regulation. *Off. J. Eur. Union* **2018**. Available online: https://gdpr-info.eu/ (accessed on 10 April 2022).
34. Yu, T.; Zhang, Y.; Lin, K.J. Modeling and Measuring Privacy Risks in QoS Web Services. In Proceedings of the 2006 IEEE International Conference on E-Commerce Technology and 2006 IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, PaloAlto, CA, USA, 26–29 June 2016.
35. Belabed, A.; Aïmeur, E.; Chikh, M.A.; Fethallah, H. A Privacy-Preserving Approach for Composite Web Service Selection. Trans. *Data Priv.* **2017**, *10*, 83–115.
36. Kuhn, H.W. The Hungarian Method for the Assignment Problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
37. Munkres, J. Algorithms for the Assignment and Transportation Problems. *SIAM J.* **1957**, *5*, 32–38. [CrossRef]
38. Ke, C.; Xiao, F.; Huang, Z.; Meng, Y.; Cao, Y. Ontology-based Privacy Data Chain Disclosure Discovery Method for Big Data. *IEEE Trans. Serv. Comput.* **2022**, *15*, 59–68. [CrossRef]
39. Shen, H.; Liu, G. An efficient and trustworthy resource sharing platform for collaborative cloud computing. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 862–875. [CrossRef]
40. Li, X.; Yuan, J.; Ma, H.; Yao, W. Fast and Parallel Trust Computing Scheme Based on Big Data Analysis for Collaboration Cloud Service. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1917–1931. [CrossRef]
41. Niwattanakul, S.; Singthongchai, J.; Naenudorn, E.; Wanapu, S. Using of Jaccard coefficient for keywords similarity. In Proceedings of the International Multi-Conference of Engineers and Computer Scientists, Hong Kong, China, 13–15 March 2013; pp. 1–5.
42. Stoneburner, G.; Goguen, A.; Feringa, A. Risk Management Guide for Information Technology Systems. *Tech. Rep.* **2002**. Available online: https://csrc.nist.gov/publications/detail/sp/800-30/archive/2002-07-01 (accessed on 10 April 2022).
43. Riaz, A.S.; Kamel, A.; Luigi, L. Dynamic Risk-Based Decision Methods for Access Control Systems. *Comput. Secur.* **2012**, *31*, 447–464.
44. Mahdi, H.; Bill, M.; Shervin, S. QoE-Aware Bandwidth Allocation for Video Traffic Using Sigmoidal Programming. *IEEE Multim.* **2017**, *24*, 80–90.
45. Phu, L.; He, Q.; Cui, G.; Xia, X.; Mohamed, A.; Feifei, C.; John, G.H.; John, G.; Yun, Y. QoE-Aware User Allocation in Edge Computing Systems with Dynamic QoS. *Future Gener. Comput. Syst.* **2020**, *112*, 684–694.
46. Zhu, H.; Zhou, M. Role Transfer Problems and Algorithms. *IEEE Trans. Syst. Man Cybern. Part A* **2008**, *38*, 1442–1450.
47. Zhu, H.; Alkins, R. Improvement to Rated Role Assignment Algorithms. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 4861–4866.
48. Zhu, H.; Zhou, M.; Alkins, R. Group Role Assignment via a Kuhn-Munkres Algorithm-Based Solution. *IEEE Trans. Syst. Man Cybern. Syst.* **2012**, *42*, 739–750. [CrossRef]
49. IBM. IBM ILOG CPLEX Optimization Studio. 2019. Available online: https://www.ibm.com/products/ilog-cplex-optimization-studio (accessed on 10 April 2022).
50. Zhu, H.; Sheng, Y.; Zhou, X.; Zhu, Y. Group Role Assignment with Cooperation and Conflict Factors. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 851–863. [CrossRef]
51. Xiao, Y.; Zhou, A.C.; Yang, X.; He, B. Privacy-Preserving Workflow Scheduling in Geo-Distributed Data Centers. *Future Gener. Comput. Syst.* **2022**, *130*, 46–58. [CrossRef]
52. Yang, M.; Yu, Y.; Bandara, A.K.; Nuseibeh, B. Adaptive Sharing for Online Social Networks: A Trade-off Between Privacy Risk and Social Benefit. In Proceedings of the 2014 IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Beijing, China, 24–26 September 2014; pp. 45–52.
53. Sourya, J.D.; Abdessamad, I. Enabling Users to Balance Social Benefit and Privacy in Online Social Networks. In Proceedings of the Annual Conference on Privacy, Security and Trust, Belfast, Northern Ireland, UK, 29 November 2018; pp. 1–10.
54. Mahmoud, B.; Charith, P.; Chirine, G.; Djamal, B. User-Centric Privacy Engineering for the Internet of Things. *IEEE Cloud Comput.* **2018**, *5*, 47–57.
55. Bikash, C.S.; Barbara, C.; Elena, F. A Risk-Benefit Driven Architecture for Personal Data Release. In Proceedings of the 2016 IEEE International Conference on Information Reuse and Integration, Pittsburgh, PA, USA, 28–30 July 2016; pp. 40–49.