Workload Factoring and Resource Sharing via Joint Vertical and Horizontal Cloud Federation Networks

Haipeng Chen, Bo An, *Member, IEEE*, Dusit Niyato, *Fellow, IEEE*,
Yengchai Soh, *Senior Member, IEEE*, and Chunyan Miao

***Abstract*—In cloud computing, a private (secondary) cloud can 1) outsource workload to public (primary) clouds via *vertical* federation or 2) share resources with other secondary clouds through *horizontal* federation to enhance its service quality. While there have been attempts to establish a joint vertical and horizontal cloud federation (VHCF), little is known regarding the economic aspects (e.g., what stable cooperation pattern will form, will it improve efficiency) of such a complex cloud network where secondary clouds are self-interested. To fill the gap, we analyze the interrelated workload factoring and federation formation among secondary clouds, while providing scalable algorithms to assist them to optimally select partners and outsource workload. We use a game theoretic approach to model the federation formation of clouds as a coalition game with externalities. We adopt a pessimistic core to characterize the cooperation stability and formulate its computation as a bilevel optimization problem. The properties of the problem are explored and efficient algorithms are developed to solve it. Experimental results show that the two common practices (no-cooperation and all-in-one federation) are not always stable. The results also show that compared with the two common practices, secondary clouds can decrease service delay penalty by around 11% with the proposed VHCF network.**

***Index Terms*—Cloud computing, cloud federation, workload factoring, coalition formation, core, bilevel optimization.**

I. Introduction

Cloud computing has emerged as a popular service model where computing resources can be accessed by users over the Internet on a usage basis, avoiding upfront investment on the physical infrastructures. It has been shown that individual private clouds (*secondary clouds*) can form *horizontal* cloud federations [1] and scale up their services by pooling their computing resources. Furthermore, the horizontally federated clouds can vertically outsource part of their requests to a public cloud (*primary cloud*) and pay the usage fee, which is called workload factoring/outsourcing. For example, in Fig. 1, if secondary cloud A receives 80 requests from its end users and has only 60 computing resource units (e.g., CPU, memory, and bandwidth), while secondary cloud B receives 20 requests and has 30 resource units, then B can share its redundant resources to process the excessive requests from A and receive a share of payoff from A, while for A, this payoff share is usually lower than the cost of outsourcing to primary clouds.

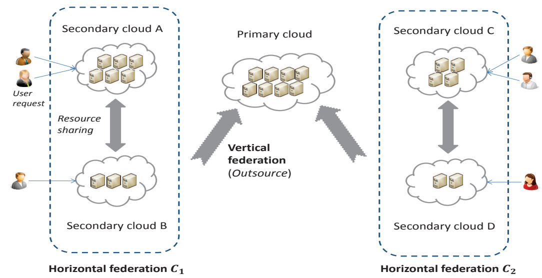


Fig. 1: A typical VHCF network. 4 secondary clouds form 2 horizontal federations. The horizontal federations can also rent resources from the primary cloud through a vertical federation.

Meanwhile, the total number of requests is still larger than the total number of resources. To finish the requests on time and avoid the penalty of violating *service level agreements (SLAs)*, the formed horizontal federation can further outsource their requests to the primary cloud via a vertical federation. Different from the resource sharing in a horizontal federation, the resources of the primary clouds are on a renting basis and are often costly. Moreover, the resources in a horizontal federation are dedicated to processing internal requests, while in a vertical federation, the resources of a primary cloud are shared by different external requests. Therefore, the performance (i.e., processing time) of the requests in a primary cloud depends on the number of external requests outsourced by other entities [2]. While horizontal federations may still experience resource shortage or redundance, vertical federation is a perfect complement. Due to the merits, such a joint *vertical and horizontal cloud federation* (VHCF) network is evaluated both theoretically [3] and practically (e.g., Aristotle federated cloud [4] and EUBrazil Cloud Connect [5]).

Although the technical aspects of both the vertical and horizontal federations have been investigated [3], [6], [7], the economic aspects of forming such a joint cloud federation have never been studied, while these individual private clouds, which can be treated as *intelligent agents* in a network, are usually self-interested. To this end, three intrinsic questions remain to be answered: 1) *Federation Formation.* If cooperation incurs a cost, what is the stable cooperative pattern? Is the grand coalition[1] - a coalition of all the clouds - always

Manuscript received May 3, 2016; revised September 30, 2016.

Haipeng Chen is with Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore (email: chen0939@e.ntu.edu.sg)

Bo An, Dusit Niyato and Chunyan Miao are with the School of Computer Science and Engineering, Nanyang Technological University, Singapre (email:boan@ntu.edu.sg, dniyato@ntu.edu.sg, ascymiao@ntu.edu.sg)

Yeng Chai Soh is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (email:ecsoh@ntu.edu.sg)

[1] We use “horizontal federation” and “coalition” interchangeably, to denote a group of cooperating secondary clouds. Note that we do not treat “vertical federation” as a coalition, since the “sharing” of resources via vertical federation is on a renting basis.

stable? How much benefit will each coalition obtain, and how to divide the benefit among the coalition members so that nobody wants to deviate? 2) *Workload factoring*. Since a coalition's task performance in a primary cloud depends on the strategies of the competing coalitions, what is the optimal workload factoring strategy of each coalition? 3) *Efficiency*. Is it beneficial for a group of secondary clouds to employ such a VHCF network?

Due to the interdependency of the competitive workload factoring among different coalitions and the cooperative resource and payoff sharing within a coalition, it is not easy to answer the aforementioned questions. Recently, several game theoretic approaches have been proposed to study the economics of cloud computing [2], [8]–[18] and other forms of networks [19]–[25]. Unfortunately, these approaches ignored either cooperation or competition among the players (see Section II for details), and thus their methods cannot be applied. We make four key contributions towards answering these challenging questions.

First, we present a new game theoretic model which considers both cooperation and competition among cloud providers in the joint VHCF network. We first establish that for a fixed coalition structure (a partition of the secondary clouds), the horizontal cloud federations play a competitive *workload factoring game*. We then propose a coalition game to study the federation formation among the secondary clouds. We show that the coalition value, which is decided by the workload factoring game, depends not only on a coalition itself, but also on the specific coalition structure. To characterize the *most stable* situation where no subset of secondary clouds want to deviate, we use the commonly adopted *least pessimistic core* as the solution concept.

Second, based on the analysis, we formulate the computation of the least pessimistic core as a bilevel optimization problem. The lower level problem computes Nash equilibrium strategies of the workload factoring game given a fixed coalition structure (it answers the second question above), and the upper level problem, using the lower level problem as a subroutine, finds the least pessimistic core and the corresponding coalition structure and payoff division (the first question).

Third, we exploit the structural properties of the game and design efficient algorithms to solve the bilevel optimization problem. We first prove the strict concavity of a coalition's value function. Using this property, we establish the existence and uniqueness of the pure strategy Nash equilibrium of the workload factoring game and significantly simplify the lower level problem. For the upper level problem which belongs to the hardest class of combinatorial optimization problems, we propose a binary search algorithm, which prunes the solution space (i.e., feasible coalition structures) based on two heuristic rules. Our proposed algorithms can assist secondary providers to optimally select partners, decide workload factoring strategies and negotiate payoff in a VHCF network.

Last, we conduct extensive experimental evaluations. We first show through a simple scenario the iterrelated workload factoring and federation formation. We then evaluate the superadditivity and externality. We also study the coalition formation results under different network conditions and show that, the two common practices - no-cooperation and the grand coalition - may not be stable if cooperation incurs a cost. We also show the improvement of the secondary clouds' total utility by using the VHCF network, comparing with no cooperation and grand coalition (it answers the last question above). Finally, we show the efficiency of the proposed algorithms in terms of runtime and optimality.

The rest of this paper is organized as follows. Related work is reviewed in Section II. Section III describes the cloud network model. Section IV presents the game theoretic analysis of the VHCF network. Section V presents algorithms for the formulated problem. Section VI provides experimental results. The paper is summarized in Section VII. We refer to Table I for a list of notations.

II. Related Work

Our work falls into the broad research area of economics of cloud computing, which can be categorized into topics including resource allocation/task scheduling [8], [9], pricing [10], [11] and workload factoring/outsourcing [2], [12], [13]. Our research belongs to the last category. There have been a few works on optimizing cloud outsourcing. In [12], the authors aim at maximizing utilization of internal data center, minimizing cost of running outsourced tasks, while maintaining service quality. In [13], workload factoring is considered in combination with the task scheduling problem within a private cloud. These works neglect the interrelated outsourcing behaviors among different secondary clouds. In [2] the authors study the workload factoring among users of the same cloud provider. A non-cooperative game is formulated where cloud users make strategic decisions on workload factoring. However, they do not consider cooperation among the cloud providers.

Competition and cooperation among different cloud providers have been independently studied in other economic aspects of cloud computing. In [14], price competition is considered in the setting of multiple cloud providers. Instead of studying the competition among cloud providers, in [15], the competition among the cloud users is investigated. These two works focus on competition among cloud providers or users, but they ignore their potential cooperation. Cooperation among clouds is studied by applying cooperative game theory. In [16] the resource and revenue sharing problem within a horizontal cloud federation is formulated as a stochastic program. In [17], a coalition game based resource allocation and revenue sharing scheme is proposed. The stable coalition structure is obtained with the merge-split algorithm. The authors in [18] introduce an MILP-based formulation of the energy-aware resource allocation within a coalition, which aims at minimizing energy cost of a coalition. However, these works only study the cooperative behavior among cloud providers.

Game theory has been extensively employed to analyze cooperation and competition in other forms of networks. In [19], the authors study the coalition formation game among small cell base stations for joint beamforming in cell networks. An overlapping coalition formation game model is proposed in [20] to study the cooperation among secondary users in cognitive radio networks, while in another work [21], the coalition formation among primary users is studied. In [22],

TABLE I: List of Notations

Symbol	Definition	β^s	penalty rate of secondary clouds
$\mathcal{I}$	the set of primary clouds	C	a coalition among secondary clouds
c_i^p	capacity of primary cloud i	ξ_k	delay function of coalition C_k
r_i^p	total number of requests received by primary cloud i	$\mathcal{C}$	the set of all possible coalitions among secondary clouds
p_i^p	price per request charged by primary cloud i	$c(C)$	capacity of coalition C
β_i^p	penalty rate of primary cloud i	$r(C)$	total number of requests received by coalition C
τ_i	delay function of primary cloud i	π	a coalition structure among secondary clouds
$\mathcal{J}$	the set of secondary clouds	Π	the set of all possible coalition structures among secondary clouds
c_j^s	capacity of secondary cloud j	z_{ki}	number of requests outsourced to primary cloud i by coalition C_k
r_j^s	number of requests received by secondary cloud j	$\mathbf{z}_k$	request strategy of coalition C_k
p_j^s	price per request charged by secondary cloud j	$\mathbf{z}$	request strategy profile of all coalitions

the authors model the cooperation of the small-scale electricity suppliers and the end users as a coalition game. In [23], the spectrum sharing among wireless users in a wireless network is modelled as a non-cooperative game. The authors in [24] study a non-cooperative resource allocation game among the device-to-device users in cell networks. In [25] a non-cooperative spectrum load balancing game among the secondary users is proposed in a cognitive radio network. These works mainly focus on resource contention among different entities.

To the best of our knowledge, no previous work has analyzed interrelated competition and cooperation among a same group of entities in cloud networks or other forms of networks. We aim at filling the gap.

III. Cloud Network System

In this section, we first introduce the secondary clouds and horizontal federation. We then describe the primary clouds and vertical federation. Last, we define the coalition values of the horizontal federations.

A. Secondary Cloud and Horizontal Federation

We consider a set of secondary clouds denoted as $\mathcal{J} = \{1, 2, \ldots, j, \ldots, |\mathcal{J}|\}$. Each secondary cloud j has a pool of computing resources. Each secondary cloud j is characterized by three parameters: the number of available resources (capacity) c_j^s, the number of requests r_j^s from its end users and the per unit utility p_j^s it gets for processing a unit user request.[2] We denote a horizontal federation, or *coalition* of a subset of secondary clouds as $C \subseteq \mathcal{J}$. Once a coalition C is formed, the resources and requests of its members are aggregated: $c(C) = \sum_{j \in C} c_j^s$ and $r(C) = \sum_{j \in C} r_j^s$. A *partition* of secondary clouds, or a *coalition structure* $\pi = \{C_1, \ldots, C_k, \ldots, C_m\}$, consists of different coalitions. We denote Π as the set of all coalition structures. Following [18] and [16], we refine that a partition π is exhaustive and disjoint: $\cup_{k=1}^m C_k = \mathcal{J}$, and $C_k \cap C_{k'} = \emptyset$ (for any $C_k, C_{k'} \in \pi$ such that $C_k \neq C_{k'}$).

B. Primary Cloud and Vertical Federation

As shown in Fig. 1, instead of processing all the requests with the federated resource pool, a horizontal federation may also outsource requests to primary clouds via *vertical federation*. We denote the set of primary clouds as $\mathcal{I} = \{1, 2, \ldots, i, \ldots, |\mathcal{I}|\}$. Similarly, each primary cloud i is characterized by its capacity c_i^p, number of requests r_i^p and unit price p_i^p. Denote the number of requests from coalition C_k to primary cloud i as z_{ki}, the total number of requests received by primary cloud i is $r_i^p = \sum_{C_k \in \pi} z_{ki}$. The *request strategy* of coalition C_k is denoted as a vector $\mathbf{z}_k = \langle z_{k1}, \ldots, z_{ki}, \ldots, z_{k|\mathcal{I}|}\rangle^T$,[3] and the *request strategy profile* associated with a coalition structure $\pi = \{C_1, C_2, \ldots, C_m\}$ is a matrix $\mathbf{z} = \langle \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\rangle$.

C. Processing Time, Delay and Coalition Value Function

Due to cloud virtualization technology, a request actually runs on a *virtual machine* (VM) instead of a physical machine. For example, c resources can host r VMs, while each VM takes up $\frac{c}{r}$ resources. By default, we assume that each request runs on one VM[4], and if one VM takes up a unit resource, its processing time is a unit period. As a result, when r requests run on c resources, the *processing time* of each request is r/c. [5] As is noticed, when $r > c$, it causes a "congestion" effect and delay of processing. According to the service level agreement [29], a cloud provider bears a *penalty* if the processing time exceeds a "deadline". By default, we set the deadline of each request as a unit time period[6]. For primary cloud i, given the total number of requests r_i^p and capacity c_i^p, its *delay* of processing time is

$$\tau_i(\mathbf{z}) = r_i^p / c_i^p - 1. \tag{1}$$

Similarly, the delay function of a coalition C_k is denoted as

$$\xi_k(\mathbf{z}) = \frac{r(C_k) - \sum_{i \in \mathcal{I}} z_{ki}}{c(C_k)} - 1, \tag{2}$$

where $r(C_k) - \sum_{i \in \mathcal{I}} z_{ki}$ is the number of requests processed in-house. Note that when capacity is larger than the total

[2]We assume r_j^s is predictable and refer to [26] as an example of demand prediction. For ease of analysis, we consider only one type of computing resource, as is the case in [2]. Throughout the paper, the superscript p means "primary", while s means "secondary".

[3]Note that this formulation is a generalization of "vendor lock-in", which means that a secondary cloud can outsource requests to only one primary cloud. Nonetheless, our formulation can be reduced to the vendor lock-in case by simply setting the number of outsourced requests to only one primary cloud as non-zero.

[4]Different request sizes can be treated as a large request being divided into smaller unit ones.

[5]we refer to [2] for a detailed discussion of processing time. Note that we do not consider requests being dropped by cloud providers. In practice, the rate of dropping a request is very low – for example, according to the service level agreement of Amazon EC2 [27] and Microsoft Azure [28], the rate of dropping a request is less than 0.05%, which is negligible.

[6]Requests with longer deadlines can be treated as several separate "virtual" requests spanning over multiple time periods. For example, if a request with a deadline of 5 unit time is received at a time t, we treat it as 5 separate virtual requests received in 5 consecutive time periods from t to $t + 4$. For each virtual request, it has a deadline of one unit time. While in the current time t, only the first among the 5 virtual requests is taken into account, and the rest 4 requests are counted as requests in future time periods.

number of requests, the delay is negative, as the requests are processed before the deadline.

We denote the penalty function of a primary cloud i as $f_i^p(\tau_i)$. For ease of analysis, we assume all secondary clouds have a same penalty function f^s. Therefore, the penalty of coalition C_k is $f^s(\xi_k)$. However, our formulation can be extended to different penalty rates[7]. Accordingly, we define the value function $v(C_k;\pi;\mathbf{z})$ of coalition $C_k \in \pi$ with request strategy $\mathbf{z}$ as:

$$v(C_k;\pi;\mathbf{z}) = \sum\nolimits_{j\in C_k} p_j^s r_j^s - \sum\nolimits_{i\in\mathcal{I}} p_i^p z_{ki} - (\sum\nolimits_{j\in C_k} r_j^s - \sum\nolimits_{i\in\mathcal{I}} z_{ki}) f^s(\xi_k(\mathbf{z})) - \sum\nolimits_{i\in\mathcal{I}} z_{ki}(f^s(\tau_i(\mathbf{z})) - f_i^p(\tau_i(\mathbf{z}))), \quad (3)$$

where the first term on the right hand side is the utility of processing requests from the end users. The second term is the payment to the primary clouds. The last two terms indicate the delay penalty for the requests processed in-house and in primary clouds, respectively. Note that for the case of negative delay, i.e., the request is finished before the deadline, the penalty can be viewed as a "bonus".

IV. Coalition Formation and Workload Factoring of Secondary Clouds

In this section, we analyze the interrelated workload factoring and coalition formation among the secondary clouds. As an example, we look at the two horizontal cloud federations in Fig. 1. First, the workload factoring among different horizontal federations is interdependent. For example, if Federation 1 reduces its outsourced request to the primary cloud, Federation 2 will have an incentive to increase its outsourced requests, since the processing time in the primary cloud is shorter[8]. Second, the workload factoring and the formation of coalition structures are interdependent. For example, if the two clouds in Federation 2 decide not to cooperate, they may need more resources and the total number of outsourced requests will increase, which further impacts the outsourcing strategy of Federation 1. On the other hand, the outsourcing strategy of Federation 1 also influences whether the rest of the clouds will cooperate. For example, if Federation 1 submits fewer requests to the primary cloud, the processing time in the primary cloud is shorter. In this case, the two clouds in Federation 2 will tend to submit more requests to the primary cloud, and it is less beneficial for them to cooperate.

Due to the above two complicated interrelations, we decompose the analysis into two steps, which is shown in Fig. 2. In the first step, we consider a fixed coalition structure and a workload factoring game. Next, we study the coalition formation among the secondary clouds and use a *least pessimistic core* to characterize the most stable outcome. Based on the analysis, we formulate the computation of the least pessimistic core as a bilevel optimization problem.

[7]When different secondary clouds have different penalty functions, they will have an incentive to allocate more requests to the clouds with smaller penalties. We can extend our formulation by incorporating a resource allocation subscheme among the clouds [8], [9].

[8]Throughout this paper, we look at small and medium scale of cloud networks, where one secondary cloud's request processing time in the primary cloud is impacted by the other secondary clouds, as is the case studied in [2].

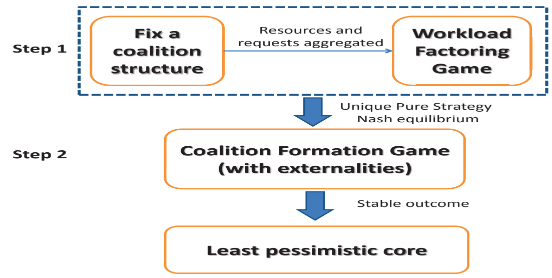


Fig. 2: Two steps to analyze the VHCF network.

A. Workload Factoring Game (WFG) for a Fixed Coalition Structure

Definition 1. *For a fixed coalition structure π, a **WFG** is a non-cooperative game, where the **players** are the different coalitions, the **strategy** is the number of requests outsourced to primary clouds, and the **utility** of a player is defined in Eq.(3).*

For non-cooperative games, Nash equilibrium (NE) is a natural solution concept, where each player is not willing to unilaterally change its strategy. Given coalition structure π, the NE strategy $\mathbf{z}_k^N$ of C_k is:

$$\mathbf{z}_k^N \in \arg\max_{\mathbf{z}_k\in\mathcal{A}_k} v(C_k;\pi;\mathbf{z}_k,\mathbf{z}_{-k}^N), \forall C_k \in \pi, \quad (4)$$

where $\mathbf{z}_{-k}^N$ denotes the equilibrium strategies of all the other coalitions. $\mathcal{A}_k$ is the feasible region of $\mathbf{z}_k$:

$$z_{ki} \geq 0, \forall i \in \mathcal{I}, \quad (5)$$

$$\sum\nolimits_{i\in\mathcal{I}} z_{ki} \leq r(C_k). \quad (6)$$

The entire feasible solution space is $\mathcal{A} = \times_{k=1}^m \mathcal{A}_k$, where $\times$ is a Cartesian product. Eq.(5) means that the number of requests from coalition C_k to each primary cloud i is non-negative, while Eq.(6) requires that the sum of requests outsourced by each coalition C_k cannot exceed its total number of requests[9].

B. Coalition Formation Game with Externalities (CFG-e)

So far, we have addressed the second question in Section I, i.e., an optimal workload factoring strategy in a competitive setting is an NE strategy. After we obtain the workload factoring strategy for each coalition structure with Eq.(4), each coalition's utility can be calculated by Eq.(3). To simplify notation, we omit $\mathbf{z}^N$, and denote the NE value of a coalition $C_k \in \pi$ as $v(C_k;\pi)$. As is noticed, the coalition value depends on the specific coalition structure. In the following, we answer the first question in Section I by proposing a coalition game model.

Definition 2. ***CFG-e** is a coalition game played by secondary clouds, where coalition value depends on the specific coalition structure, i.e., it has **externality** effect. The **outcome** of a coalition game is a tuple $(\pi,\mathbf{x})$, where $\mathbf{x} = \langle x_1,\ldots,x_j,\ldots,x_{|\mathcal{J}|}\rangle$ denotes the payoff of each secondary cloud.*

Since the secondary clouds are self-interested, we need to consider a *stable* outcome, i.e., no subset of players can get

[9]We leave the description of solution algorithms of the WFG and the following CFG-e in Section V.

a higher payoff by deviating from the current outcome and forming a new coalition. However, for a CFG-e, when the players want to deviate, they have to consider the response of the *residual players* (i.e., the rest of the players). There has been extensive discussion on the stability in a CFG-e. We refer to Koczy et al. [30] for a review. In this paper, we use the commonly adopted *pessimistic core*[10] as the solution concept, which means that if a group of players want to deviate, they are pessimistic about their potential coalition value [32].

Definition 3. *The* ***pessimistic value*** *of a coalition C in a CFG-e is the lowest possible value in all coalition structures Π_C which contain C: $\tilde{v}(C) = \min_{\pi \in \Pi_C} v(C, \pi)$.*[11]

Consider an outcome $(\pi, \mathbf{x})$, where $x_j \geq 0, \forall j \in \mathcal{J}$, it is in the *pessimistic core* if $\sum_{j \in C} x_j \geq \tilde{v}(C),\ \forall C \in \mathcal{C}$, and $\sum_{j \in C} x_j = v(C, \pi),\ \forall C \in \pi$. The two constraints ensure that 1) the sum of payoff for each subset of players is no smaller than the pessimistic value of their intended coalition and 2) the sum of payoff of all the coalition members is equal to the coalition value. However, the pessimistic core can be empty, or there can be multiple cores. Thus we define a "least pessimistic core" to denote the set of most stable outcomes for a CFG-e.

Definition 4. *The* ***least pessimistic core*** *of a CFG-e is the outcome $(\pi, \mathbf{x})$ with the smallest ε satisfying*

$$\sum\nolimits_{j \in C} x_j \geq \min\nolimits_{\pi \in \Pi_C} v(C, \pi) - \varepsilon,\ \forall C \in \mathcal{C} \quad (7)$$

$$\sum\nolimits_{j \in C} x_j = v(C, \pi),\ \forall C \in \pi \quad (8)$$

$$x_j \geq 0, \forall j \in \mathcal{J} \quad (9)$$

When $\varepsilon > 0$, a set of players prefer to deviate when the loss of deviation is no larger than ε, while $\varepsilon \leq 0$ means that they prefer to stay at the current outcome even if they can get a reward of $-\varepsilon$ if they deviate. Therefore, the outcome with the smallest ε is the most stable. Note that in Eq.(7), to obtain the pessimistic value of a coalition C, we need to go through all the coalition structures that contain C, which is impractical when the scale of the problem is large. Before we finally formulate the bilevel optimization problem to obtain the least pessimistic core and the corresponding outcome, we simplify Eq.(7) with two important properties of the proposed CFG-e, namely, *superadditivity* and *positive externality*.

Lemma 1. *When cooperation incurs no cost, the proposed CFG-e is* ***superadditive****, i.e., for any two coalitions C_k, $C_{k'}$, there is $v(C_k \cup C_{k'}; \{C_k \cup C_{k'}\} \cup \pi_r) \geq v(C_k; \{C_k, C_{k'}\} \cup \pi_r) + v(C_{k'}; \{C_k, C_{k'}\} \cup \pi_r)$, where π_r denotes a partition of the residuals $\mathcal{J} \setminus (C_k \cup C_{k'})$. Coalition structure $\{C_k \cup C_{k'}\} \cup \pi_r$ means that C_k and $C_{k'}$ cooperate, and $\{C_k, C_{k'}\} \cup \pi_r$ means that they do no cooperate.*

Proof. The proof is in two phases. First, we prove that for two coalitions C_k and $C_{k'}$, if the total number of their outsourced requests after cooperation equals the sum of outsourced requests before cooperation, their total delay is no larger after cooperation. Second, we prove that, if the total number outsourced requests can be changed, the total delay (utility) is still no larger (smaller) after cooperation.

1) *Total number of outsourced requests do not change.* We first assume that after C_k and $C_{k'}$ form a new coalition, the total number of outsourced requests equals the sum of their outsourced requests before cooperation. In this case, only the delay penalty (the third term in Eq.(3)) in the coalition value function is changed. Denote the number of requests processed in-house as $r_{in}(C_k) = r(C_k) - \sum_{i \in \mathcal{I}} z_{ki}$ and $r_{in}(C_{k'}) = r(C_{k'}) - \sum_{i \in \mathcal{I}} z_{k'i}$, we can obtain the total delay of C_k and $C_{k'}$ under two cases: 1) C_k and $C_{k'}$ cooperate and form a new coalition; 2) C_k and $C_{k'}$ do not cooperate. By subtracting the total delay of these two cases, we have:

$$r_{in}(C_k)(\frac{r_{in}(C_k)}{c(C_k)} - 1) + r_{in}(C_{k'})(\frac{r_{in}(C_{k'})}{c(C_{k'})} - 1) - (r_{in}(C_k) + r_{in}(C_{k'}))$$
$$(\frac{r_{in}(C_k) + r_{in}(C_{k'})}{c(C_k) + c(C_{k'})} - 1) = \frac{r_{in}^2(C_k)}{c(C_k)} + \frac{r_{in}^2(C_{k'})}{c(C_{k'})} - \frac{(r_{in}(C_k) + r_{in}(C_{k'}))^2}{c(C_k) + c(C_{k'})}$$
$$= \frac{(r_{in}(C_k)c(C_{k'}) - r_{in}(C_{k'})c(C_k))^2}{c(C_k)c(C_{k'})(c(C_k) + c(C_{k'}))} \geq 0.$$

The equality holds (i.e., the total delays of the above two cases are equal) when the processing rates of coalitions C_k and $C_{k'}$ are the same (i.e., $\frac{r_{in}(C_k)}{c(C_k)} = \frac{r_{in}(C_{k'})}{c(C_{k'})}$). Therefore, the total delay (also for unit delay, since the total number of requests is the same whether the two coalitions cooperate or not) is always no larger when clouds cooperate than the total delay when they do not cooperate.

2) *Total number of outsourced requests can be changed.* If the new coalition $C_k \cup C_{k'}$ can change the number of outsourced requests, it can at least maintain the current request strategy (or change it only when it is beneficial). As a result, the total delay of the merged new coalition is still always no larger than the total delay of the two separate coalitions. Correspondingly, the utility of the merged new coalition is always no smaller than the sum of utility of C_k and $C_{k'}$ if these two coalitions do not cooperate. □

Lemma 2. *When cooperation incurs no cost, the proposed CFG-e has* ***positive externality****, i.e., for any coalitions $C_k, C_{k'}, C_{k''} \subseteq \mathcal{J}$, and a partition π_r of the residuals $\mathcal{J} \setminus (C_k \cup C_{k'} \cup C_{k''})$, the value of C_k is always no smaller when $C_{k'}$ and $C_{k''}$ cooperate: $v(C_k; \{C_k, C_{k'} \cup C_{k''}\} \cup \pi_r) \geq v(C_k; \{C_k, C_{k'}, C_{k''}\} \cup \pi_r)$.*

Proof Sketch. At the NE strategy of a coalition, if the coalition increases its outsourced requests by a very small amount (it brings no change of the processing time in either the coalition or the primary clouds), the benefit from the decrease of delay penalty in-house equals the payment to the primary clouds plus the delay penalty from the primary clouds. According to the proof of Lemma 1, when the new coalition $C_k \cup C_{k'}$ maintains its request strategy, its unit delay never increases. Thus the new coalition has an incentive to decrease its outsourced request so that the delay in-house increases, the delay in primary clouds decreases, and the equilibrium is reached again. When the delay in primary clouds decreases, the delay penalty of other coalitions decreases, and their utility increases. □

[10] We use core instead of Shapley value [31] because to incentivize the self-interested clouds, stability is more desired than fairness.

[11] Our framework can be easily extended to the optimistic core [30] by defining an optimistic value $\hat{v}(C) = \max_{\pi \in \Pi_C} v(C, \pi)$.

Theorem 1. *When cooperation incurs no cost, the grand coalition is the most stable coalition structure.*

Proof Sketch. When cooperation incurs no cost, superadditivity implies that cooperation is beneficial to the coalition members, while positive externality implies that cooperation is beneficial to the clouds outside the coalition. Thus the grand coalition will always be the most stable coalition structure. □

However, forming coalitions incurs additional cost, such as extra hypervisor machines, administration cost and data transmission cost. Therefore, we add a cost term to the coalition value function:

$$v'(C_k, \pi) = v(C_k, \pi) - C^{co}(C_k). \quad (10)$$

Since the coalition cost is generated in the intra communication within a coalition, we assume it depends only on the coalition itself, i.e., it does not have externality effect.

Theorem 2. *When coalition cost has no externality effect, the proposed CFG-e has positive externalities.*

Proof Sketch. This is a direct extension of Lemma 2. □

According to Theorem 2, the pessimistic value of a coalition C is associated with the coalition structure π where all residual players form *singleton coalitions*, i.e., each coalition has only one member. With this property, the pessimistic value of all coalitions can be obtained directly, which greatly simplifies Eq.(7). Combining this property with Eqs.(4), (7)-(9), the solution (i.e., the most stable outcome) of this CFG-e is expressed with the following bilevel optimization problem:

$$\min_{\pi,\mathbf{x},\mathbf{z}^N} \quad \varepsilon \quad (11)$$

$$\text{s.t.} \quad \sum_{j\in C} x_j \geq \tilde{v}(C) - \varepsilon, \ \forall C \in \mathcal{C} \quad (12)$$

$$\sum_{j\in C_k} x_j = v'(C_k; \pi; \mathbf{z}^N), \forall C_k \in \pi \quad (13)$$

$$x_j \geq 0, \ \forall j \in \mathcal{J} \quad (14)$$

$$\pi \in \Pi \quad (15)$$

$$\mathbf{z}_k^N \in \operatorname{argmax}_{\mathbf{z}_k\in\mathcal{A}_k} v(C_k; \pi; \mathbf{z}_k, \mathbf{z}_{-k}^N), \forall C_k \in \pi \quad (16)$$

Eqs.(11)-(15) describe the upper level problem of computing the pessimistic core, while Eq.(16) denotes the lower level problem of computing an NE strategy for a fixed coalition structure π. Eqs.(11)-(14) compute the least ε value for a fixed π. Eq.(15) denotes the feasible coalition structure space, which will be introduced in detail in Section V.

V. Computing Least Pessimistic Core

Due to the two-level structure and involved form of coalition value functions, there is no existing method to solve the above optimization problem directly. In this section, we decompose the bilevel problem and design algorithms for both levels. In the lower level, we propose two algorithms CLONE and CLOSE. CLONE employs the constraint generation method, and is a generic algorithm. CLOSE is a more efficient algorithm which employs several structural properties of the lower level problem, but is limited to concave coalition value functions. In the upper level, we propose Brute-Force and CUBE. CUBE implements a binary search method with two heuristic rules and is more efficient than Brute-Force.

A. Solving the Lower Level WFG

Given a coalition structure π, the lower level problem in Eq.(16) is a feasibility problem:

$$\max_{\mathbf{z}^N} 0 \quad (17)$$

$$v(C_k; \pi; \mathbf{z}^N) \geq v(C_k; \pi; \mathbf{z}_k, \mathbf{z}_{-k}^N), \forall \mathbf{z}_k \in \mathcal{A}_k, C_k \in \pi \quad (18)$$

$$\mathbf{z}^N \in \mathcal{A} \quad (19)$$

The General Algorithm CLONE: Unfortunately, the number of constraints in Eq.(18) is infinite since $\mathbf{z}_k \in \mathcal{A}_k$ is a continuous parameter. We use the constraint generation method [33] to deal with the infinite constraint space. The basic idea is to start with solving a relaxation of the original problem by sampling a small subset of constraints and then use a sub-routine to find violated constraints, and gradually expand the set by adding the violated constraints to the relaxed problem until no violated constraint is found. CLONE (Compute Lower level Optimization with coNstraint gEneration), as shown in Algorithm 1, begins by randomly sampling a finite subset Ω_k of constraints from $\mathcal{A}_k$ for each coalition C_k. A relaxed *master* problem is solved which is in the same form as the original feasibility problem in Eqs.(17)-(19) and returns the feasible solution $\mathbf{z}^*$. The only difference is that the master problem replaces $\mathcal{A}_k$ with Ω_k in Eq.(18). Constraint generation is then applied to find the "mostly violated" constraint for each coalition C_k, as is shown in the following *slave* problem:

Algorithm 1: CLONE

```
Sample a subset Ω of constraints from A;
repeat
    z^N ← solution of master problem;
    z_k^opt ← solution of slave problem,
    Df_k^opt ← objective of slave problem;
    for C_k ∈ π do
        if Df_k^opt > e_k then add z_k^opt to Ω;
until Df_k^opt ≤ e_k, ∀C_k ∈ π;
return z_k^N, v(C_k; π; z^N), ∀C_k ∈ π.
```

$$\max_{\mathbf{z}_k\in\mathcal{A}_k} \ Df_k = v(C_k; \pi; \mathbf{z}_k, \mathbf{z}_{-k}^*) - v(C_k; \pi; \mathbf{z}_k^*, \mathbf{z}_{-k}^*). \quad (20)$$

The most violated constraint refers to the parameter $\mathbf{z}_k$ that maximizes the difference between the utility of coalition C_k computed by $\mathbf{z}_k$ and by the optimal solution $\mathbf{z}_k^*$ of the master problem. The solutions of the slave problem return the set of mostly violated constraints, which will be added to the set of constraints Ω of the master problem. The process iterates until convergence, i.e., the optimal value Df_k^{opt} of the slave problem corresponding to each coalition $C_k \in \pi$ is no larger than a small threshold e_k. This threshold is set as the multiplication of the coalition's gross revenue $\sum_{j\in C_k} r_j^s p_j^s$ and a small ratio e (we will evaluate the runtime of the algorithm with different scales of e in Section VI). In this way, we decompose the original optimization problem into a series of master-slave non-linear optimization problems, which have a finite number of constraints and can be solved directly by existing optimization solvers like KNITRO.

The Faster Algorithm CLOSE: With CLONE, we can solve the lower level optimization with any form of coalition value functions. However, it solves a series of non-linear master-slave problems, and thus becomes rather slow for large scale settings[12]. In the following, we first demonstrate that under the commonly employed linear delay penalty rules [34]–[36] (i.e., $f_i^p(\tau_i) = \beta_i^p \tau_i$, $f^s(\xi_k) = \beta^s \xi_k$, where β_i^p and β^s are penalty rates for primary cloud i and secondary clouds, resp.), the value function of each coalition C_k is strictly concave w.r.t. its request strategy vector $\mathbf{z}_k$. We then use this property to establish the existence and uniqueness of the NE. Based on the theoretical analysis, we introduce a more efficient algorithm to solve the lower level problem.

Algorithm 2: CLOSE

```
z* ← solution of Eq.(21);
for C_k ∈ π do
    for i ∈ I do
        if z*_ki < 0 then
            Replace the corresponding equation in Eq.(21) by
            z_ki = 0;
z^N ← Solution of the new equation system;
return z^N_k, v(C_k; π; z^N), ∀C_k ∈ π.
```

Lemma 3. *With linear penalty functions $f_i^p(\tau_i) = \beta_i^p \tau_i$, $f^s(\xi_k) = \beta^s \xi_k$, the coalition value function defined in Eq.*(3) *is strictly concave w.r.t. its request strategy $\mathbf{z}_k$.*

Proof. Let $A_k = \frac{\beta^s}{c(C_k)}$, $D_k = \sum_{j \in C_k} p_j^s r_j^s - \beta^s r(C_k)(\frac{r(C_k)}{c(C_k)} - 1)$, $v_k = v(C_k; \pi; \mathbf{z})$, we rewrite the value function in Eq.(3) as: $v_k = -A_k(\sum_{i \in \mathcal{I}} z_{ki})^2 - \sum_{i \in \mathcal{I}}(\frac{(\beta^s - \beta_i^p) z_{ki}}{c_i^p} \sum_{C_{k'} \in \pi} z_{k'i}) + (\frac{2\beta^s r(C_k)}{c(C_k)} - \beta_i^p) \sum_{i \in \mathcal{I}} z_{ki} + D_k - \sum_{i \in \mathcal{I}} p_i^p z_{ki}$. The Hessian matrix of $-v_k$ is defined as $H_{ll'}(-v_k) = \frac{\partial^2 v_k}{\partial z_{kl} \partial z_{kl'}}$, $l, l' \in \mathcal{I}$. Substitute v_k with the above equation, we have: $H_{ll'}(-v_k) = \begin{cases} 2(A_k + \frac{(\beta^s - \beta_i^p) p_i^p}{c_l^p}), & l = l', \\ 2A_k, & l \neq l'. \end{cases}$ Let $B_l = \frac{\beta^s - \beta_i^p}{c_l^p}$. For any upper left $\tilde{l} \times \tilde{l}$ matrix, where $\tilde{l} = 1, \ldots, i, \ldots, |\mathcal{I}|$, its determinant

$$detH_{\tilde{l}} = 2^{\tilde{l}} \begin{vmatrix} A_k + B_1 & A_k & \cdots & A_k \\ A_k & A_k + B_2 & \cdots & A_k \\ \cdots & \cdots & \cdots & \cdots \\ A_k & A_k & \cdots & A_k + B_{\tilde{l}} \end{vmatrix} = 2^{\tilde{l}} \begin{vmatrix} A_k + B_1 & A_k & \cdots & \cdots & A_k \\ -B_1 & B_2 & 0 & \cdots & 0 \\ \cdots & 0 & B_3 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ -B_1 & 0 & \cdots & 0 & B_{\tilde{l}} \end{vmatrix}$$

$$= 2^{\tilde{l}} \begin{vmatrix} A_k + B_1 + A_k \sum_{l=2}^{\tilde{l}} \frac{B_1}{B_l} & A_k & \cdots & \cdots & A_k \\ 0 & B_2 & 0 & \cdots & 0 \\ \cdots & 0 & B_3 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 & B_{\tilde{l}} \end{vmatrix} = (A_k + B_1 + A_k \sum_{l=2}^{\tilde{l}} B_1/B_l) \Pi_{l=2}^{\tilde{l}} B_l,$$

Since $A_k > 0, B_l > 0$, then $detH_{\tilde{l}} > 0, \forall \tilde{l} \in \mathcal{I}$, i.e., the Hessian matrix of $-v_k$ w.r.t. $\mathbf{z}_k$ is positive definite everywhere. Thus $-v_k$ is strictly convex w.r.t. $\mathbf{z}_k$, while $v(C_k; \pi; \mathbf{z}_k)$ is strictly concave. □

The strict concavity of coalition function is intuitive: when the number of outsourced requests is small, the coalition value increases with the outsouced requests because the in-house overload is relieved. When the number of outsourced requests is too large, the coalition value begins to decrease because the benefit of relieving in-house overload is smaller than the cost of outsourcing.

[12]CLONE is still meaningful because when coalitions' value functions are not strictly concave, only CLONE can be applied.

Theorem 3. ***(Existence and uniqueness of equilibrium)*** *With linear delay penalty, the pure strategy Nash equilibrium for the WFG given a coalition structure π is non-empty and unique.*

Proof. For each coalition C_k as a player of the WFG, given the strategies $\mathbf{z}_{-k}$ of the other players, it will select the optimal strategy $\mathbf{z}_k$ that maximizes its value, which is a mapping $\phi_k : \mathcal{A}_{-k} \to \mathcal{A}_k$ defined as $\mathbf{z}_k = \phi_k(\mathbf{z}_{-k})$. According to Lemma 3, the value function $v(C_k; \pi; \mathbf{z})$ of each coalition C_k is strictly concave w.r.t. $\mathbf{z}_k$. Combined with the compactness (closed and bounded) of both the domain $\mathcal{A}_{-k}$ and the range $\mathcal{A}_k$, we have that the mapping $\phi_k(\mathbf{z}_{-k})$ is non-empty and unique. Also, the compactness of $\mathcal{A}_{-k}$ implies that the mapping $\phi_k : \mathcal{A}_{-k} \to \mathcal{A}_k$ is upper hemicontinuous. The non-emptiness, uniqueness (point to point), and upper hemicontinuity are naturally inherited by the mapping $\phi : \mathcal{A} \to \mathcal{A}$ defined as $\phi(\mathbf{z}) = \times_k^m \phi_k(\mathbf{z}_{-k})$. Moreover, since the vertices of $\mathcal{A}$ defined in Eqs.(5)-(6) are affinely independent, $\mathcal{A}$ is a simplex (and closed). Applying Kakutani's fixed point theorem [37], there exists a unique $\mathbf{z}^N \in \mathcal{A}$ such that $\mathbf{z} = \phi(\mathbf{z})$. Thus, the pure strategy NE of our problem is non-empty and unique. □

Based on Theorem 3, we design a more efficient algorithm for the lower level problem.

Theorem 4. *When the feasible region $\mathcal{A}$ in Eq.*(19) *is relaxed to $\mathbb{R}^{m \times n}$, the lower level optimization problem is equivalent to solving the following linear equation system:*

$$\frac{\partial v(C_k; \pi; \mathbf{z})}{\partial z_{ki}} = 0, \quad \forall i \in \mathcal{I}, \quad \forall C_k \in \pi. \tag{21}$$

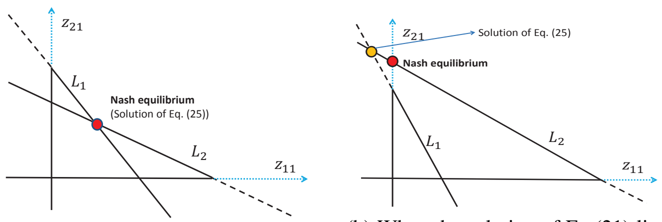
Proof. When the feasible region is relaxed to $\mathbf{z} \in \mathbb{R}^{m \times n}$, then the computation of the NE for a fixed coalition structure π is equivalent to: $\max_{\mathbf{z}_k} \ v(C_k; \pi; \mathbf{z}), \forall C_k \in \pi$. According to Lemma 3, when the request strategy $\mathbf{z}_{-k}$ of the other coalitions is fixed, the value function of each coalition is strictly concave w.r.t. $\mathbf{z}_k$. Employing the first order condition, for coalition C_k, its unique optimal strategy vector can be computed as: $\frac{\partial v_k}{\partial z_{ki}} = 0, \forall i \in \mathcal{I}$. Considering all the coalitions, we obtain Eq.(21). □

Unfortunately, the solution of Eq.(21) may not always lie in the feasible region in Eqs.(5)-(6).

Theorem 5. *The solution of Eq.*(21) *is guaranteed to satisfy the feasibility constraints in Eq.*(6)*.*

Proof. With Theorem 4, for each coalition $C_k \in \pi$, we compute its NE strategy vector with $\frac{2\beta^s}{c(C_k)}(r(C_k) - \sum_{i \in \mathcal{I}} z_{ki}) - \frac{\beta^s - \beta_i^p}{c_i^p}(\sum_{C_{k'} \in \pi} z_{k'i} + z_{ki}) - p_i^p - \beta_i^p = 0$. Extract $r(C_k) - \sum_{i \in \mathcal{I}} z_{ki}$, we have $r(C_k) - \sum_{i \in \mathcal{I}} z_{ki} = \frac{c(C_k)}{2\beta^s}(\frac{\beta^s - \beta_i^p}{c_i^p}(\sum_{C_{k'} \in \pi} z_{k'i} + z_{ki}) + p_i^p)$. Practically, the price of primary clouds is much lower than that of secondary clouds. Thus, we assume their penalty rate is also lower than that of secondary clouds, i.e., $\beta^s > \beta_i^p$. Moreover, we have $z_{ki} \geq 0$, and thus $\sum_{i \in \mathcal{I}} z_{ki} \leq r(C_k)$ is satisfied. □

Theorem 5 implies that the feasibility constraint in Eq.(6) can be omitted. In the next, we further discuss the feasibility constraint in Eq.(5). As we know, each partial differential equation in Eq.(21) can be viewed as a *best response curve*



(a) When the solution of Eq.(21) lies inside $\mathcal{A}$.

(b) When the solution of Eq.(21) lies outside $\mathcal{A}$.

Fig. 3: Two cases of NE. x and y axes denote the request strategy of coalitions C_1 and C_2, resp. The two lines L_1 and L_2 denote the *best responses curves* of coalitions C_1 and C_2 when the feasible region $\mathcal{A}$ is relaxed, while the dashed line segments are outside the feasible region. Following Theorem 6, they are replaced by the dotted line segments.

of a coalition given the strategies of other coalitions (Fig. 3). The intersection of these curves is the solution of Eq.(21). However, for each curve, there is a segment that lies outside the feasible region (the dashed line segments in Fig. 3).

Theorem 6. *For the best response curves of a coalition described in Eq.*(21)*, the segments that lie outside the feasible region (i.e.,* $z_{ki} < 0$*) should be replaced by the curve* $z_{ki} = 0$.

Proof. According to the definition of strict concavity, for the best response curve where $z_{ki} < 0$, we have $v(C_k; \pi; z''_{ki} = 0, \mathbf{z}_{k,-i}, \mathbf{z}_{-k}) > \lambda v(C_k; \pi; z_{ki} < 0, \mathbf{z}_{k,-i}, \mathbf{z}_{-k}) + (1 - \lambda) v(C_k; \pi; z'_{ki} > 0, \mathbf{z}_{k,-i}, \mathbf{z}_k)$, where $\mathbf{z}_{k,-i}$ is the request strategy to primary clouds except i, z_{ki} is on the best response curve, $z'_{ki} > 0$ is any value in the feasible region $\mathcal{A}$. Since $z''_{ki} = 0$ is between z_{ki} and z'_{ki}, then $0 < \lambda < 1$. Also we have $v(C_k; \pi; z''_{ki} = 0, \mathbf{z}_{k,-i}, \mathbf{z}_{-k}) < v(C_k; \pi; z_{ki} < 0, \mathbf{z}_{k,-i}, \mathbf{z}_{-k})$. Therefore, $v(C_k; \pi; z''_{ki} = 0, \mathbf{z}_{k,-i}, \mathbf{z}_{-k}) > v(C_k; \pi; z'_{ki} > 0, \mathbf{z}_{k,-i}, \mathbf{z}_{-k})$ holds for any $z'_{ki} > 0$ and the best response curve should be $z_{ki} = 0$. □

Based on the properties above, we propose CLOSE (Compute Lower level Optimization with Strictly concavE value function) to solve the lower level problem in Eqs.(17)-(19), as shown in Algorithm 2.

B. Solving the Upper Level CFG-e

Since the coalition value function is in an implicit form which depends on the solution of the lower level problem, the upper level problem cannot be formulated as a single integer-linear program. Instead, a separate linear program (Eqs.(11)-(14)) has to be solved to find the least core of each coalition structure. A natural idea is to do a brute-force search within Π to find the least pessimistic core.

The Exact Algorithm Brute-Force: We first introduce a *coalition structure graph*.

Definition 5. *A* ***coalition structure graph (CSG)*** Π *is a graph with* $|\mathcal{J}|$ *levels, where each node denotes a coalition structure, and each* ***level*** $\Pi_K, K = 1, 2, ..., |\mathcal{J}|$ *contains the coalition structures that have* K *coalitions. Fig. 4 shows an example CSG with* 3 *secondary clouds.*

Algorithm 3: Brute-Force

```
ε* ← M;
for  Π_K, K = 1 : |J| do
    ε*_K ← M;
    for node π ∈ Π_K do
        Solve Eq.(16) for π, get coalition values;
        ε_π ← solution of Eqs.(11)-(14);
        if ε*_K > ε_π then ε*_K ← ε_π
    if ε* > ε*_K then ε* ← ε*_K
return ε*
```

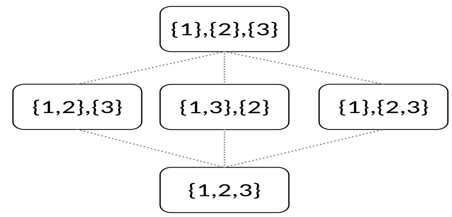


Fig. 4: A CSG with 3 secondary clouds. The bottom node (level 1) is the grand coalition, the top node (level 3) has 3 singleton coalitions, and all nodes in level 2 have 2 coalitions.

A brute-force searching algorithm, as shown in Algorithm 3, traverses all the nodes from bottom-up, level by level. For each node, a linear program (Eqs.(11)-(14)) is called as a subroutine to obtain the least core value ε_π for the node. ε^* and ε^*_K respectively denote the least core value for Π and in the level Π_K, which are initialized as a very large value M.

The Heuristic Algorithm CUBE: Note that the number of coalition structures is a *Bell number* w.r.t. the number of secondary clouds n, which belongs to one of the most notorious large numbers in the combinatorial mathematics and is of order $O(n^n)$ and $\omega(n^{n/2})$ [38]. As a result, it is rather inefficient to do a brute-force search within the exponentially large solution space. In fact, finding the least core for large scale coalition games with externalities, which belongs to hardest class of *combinatorial optimization* problems, has long been an open problem. Although there have been attempts to solve much simpler problems [39], [40], no attempts have been made to obtain the least core for large scale coalition games with externalities. In this paper, we design a binary search algorithm, CUBE, to solve the upper level problem, which implements two heuristic pruning rules based on the properties of the CFG-e.

Heuristic Rule 1: *When* $\varepsilon^*_K < \varepsilon^*_{K'}$*, search in levels closer to* Π_K. $\varepsilon^*_K < \varepsilon^*_{K'}$ implies that the nodes in Π_K tend to be more stable than those in $\Pi_{K'}$. Thus we put more search effort to the levels closer to Π_K.

Heuristic Rule 2: *For a coalition* $C \in \pi$*, if the ratio of its request and capacity* $\frac{r(C)}{c(C)}$ *satisfies* $\frac{r(C)}{c(C)} < 1$ *or* $\frac{r(C)}{c(C)} > \theta \times \frac{\sum_{C \in \pi} r(C)}{\sum_{C \in \pi} c(C)}$ [13]*, prune this node* π. The intuition behind this heuristic rule is that when a coalition C has redundant resources ($\frac{r(C)}{c(C)} < 1$) or has too low available resources ($\frac{r(C)}{c(C)} > \theta \times \frac{\sum_{C \in \pi} r(C)}{\sum_{C \in \pi} c(C)}$), the coalition value is expected to be low and thus the coalition will not form. If any

[13] It reveals the average request-capacity ratio of all secondary clouds.

such coalition C exists, a coalition structure which contains C cannot exist either. With smaller values of θ, the pruning is more aggressive, while sacrificing optimality.

Based on the above two rules, we propose CUBE (Compute Upper level problem with Binary sEearch). As shown in Algorithm 4, the algorithm iteratively searches within the smallest level Π_K to the largest level $\Pi_{K'}$, which are initialized as 1 and $|\mathcal{J}|$, respectively. For $K < K'$, meaning that there are more than 2 levels of nodes, CUBE computes the least core $\varepsilon_K^*, \varepsilon_{K'}^*$ of these two levels (Lines 4-11). In Lines 7-8, it skips any node which satisfies *heuristic rule 2*. If $\varepsilon_K^* < \varepsilon_{K'}^*$, according to *heuristic rule 1*, the largest level becomes $K' = \lfloor (K+K')/2 \rfloor$, which means that half of the levels closer to level $\Pi_{K'}$ are pruned. Else, the smallest level is updated as $K \leftarrow \lceil (K+K')/2 \rceil$.

Algorithm 4: CUBE

```
1  ε* ← M; K ← 1, K' ← |J|;
2  while K < K' do
3      ε*_K ← M, ε*_K' ← M;
4      for node π ∈ Π_K do
5          for C ∈ π do
6              if r(C)/c(C) ≥ 1 && r(C)/c(C) ≤ T_2 then
7                  Solve Eq.(16) for π, get coalition values;
8                  ε_π ← solution of Eqs.(11)-(14);
9                  if ε*_K > ε_π then ε*_K ← ε_π;
10     Repeat Lines 5-10 for Π_K', get ε*_K';
11     if ε*_K < ε*_K' then K' ← ⌊(K + K')/2⌋;
12     else K ← ⌈(K + K')/2⌉;
13 return ε*;
```

VI. Experimental Results

In this section, we 1) show through a simple example the interrelated workload factoring and coalition formation, and the general process of obtaining the least pessimistic core; 2) evaluate superadditivity and externality; 3) evaluate coalition formation under different VHCF scenarios; 4) show the improvement of total utility by using the VHCF network; 5) evaluate the performance of different algorithms.

A. Obtaining Least Pessimistic Core: A Simple Scenario

To show the interrelated workload factoring and coalition formation, and the general process of obtaining the least pessimistic core, we first study a simple scenario of 1 primary cloud and 3 secondary clouds. Note that in this subsection, we do not intend to evaluate how different parameters (e.g., the price, penalty of secondary clouds, coalition cost, demand and supply sizes. We refer to Table II for a summary of parameters that are mentioned in the experimental settings.) influence the coalition formation results among the secondary clouds (which will be evaluated in Section VI-C). As a result, in this subsection, we just assign arbitrary values to these parameters, as described in the following.

By default, we set the price of the primary cloud as 1. Correspondingly, we set the price vector of the 3 secondary clouds as $\langle 2, 1.98, 2.01 \rangle$ — approximately twice as that of the primary cloud's. The prices of the secondary clouds are set close because, to ensure interoperability among different secondary clouds, the application service in these cooperating clouds should be similar, while the prices of their service should also be close. The capacity of the primary cloud is set as 40. The capacity and request vectors of the 3 secondary clouds are $\langle 40, 50, 60 \rangle$ and $\langle 60, 40, 90 \rangle$, resp. Following the line of research of [34]–[36], throughout this section, we adopt the linear penalty rules. In this subsection, the penalty rates of the primary clouds and secondary clouds are set as $\beta^p = 1$ and $\beta^s = 2$, resp. Note that when penalty rate equals price, it means if processing delay is twice as much as the deadline, the penalty for delay equals the price paid and the provider gets no payment from the user.

TABLE II: Summary of parameters mentioned in the experimental settings. We evaluate the influence of these parameter values on coalition formation results in Section VI-C.

Parameter	Description
p_i^p	price per request of primary cloud i
c_i^p	capacity of primary cloud i
β_i^p	penalty rate of primary cloud i
p_j^s	price per request of secondary cloud j
c_j^s	capacity of secondary cloud j
r_j^s	number of requests of secondary cloud j
β^s	penalty rate of secondary clouds
c^{co}	unit coalition cost
γ	coalition size impact factor

Coalition cost is defined as[14]: $C^{co}(C_k) = c^{co}(|C_k| - 1)^\gamma$, where c^{co} is the unit coalition cost. $(|C_k| - 1)^\gamma$ is determined by the coalition size $|C_k|$. For example, when a coalition is a singleton, there is no coalition cost. Whenever a new member enters the coalition, it incurs additional coalition cost. γ is a factor which measures the extent to which coalition size influences the coalition cost. Larger γ values indicate larger coalition size effect. We set $\gamma = 1$ (i.e., coalition cost is linear w.r.t. coalition size), $c^{co} = 6$ (i.e., when a new member joins the coalition, an additional cost of 6 is produced).

First, we call CLONE/CLOSE to compute the NE strategies and coalition values of different coalition structures, which are shown in the 2^{nd} and 3^{rd} columns of Table III. We can see that the workload factoring and the coalition formation are interdependent. For example, the request strategy of coalition $\{1\}$ is 29.4 when secondary clouds 2 and 3 cooperate, while 28.5 when they do not cooperate. Alternatively, it can be viewed as the request strategy of $\{1\}$ influences whether secondary clouds 2 and 3 cooperate. We can also see that the coalition value of a coalition depends on the coalition structure, i.e., the externality effect, which will be studied in detail in the next subsection.

After we obtain the coalition values for each coalition structure, we can get the pessimistic value of each coalition. For example, the two coalition structures which contain coalition $\{1\}$ are $\{1\},\{2,3\}$ and $\{1\},\{2\},\{3\}$. According to Definition 3, the pessimistic value of $\{1\}$ is the smaller one among the two values 130.1 and 125.0. Therefore, the pessimistic value of coalition $\{1\}$ is $\tilde{v}(\{1\}) = 125.0$. In fact, this is the direct result of Theorem 2, since in coalition structure $\{1\},\{2\},\{3\}$, both $\{2\}$ and $\{3\}$ are singleton coalitions. Similarly, we can obtain the pessimistic value of all the coalitions in Table IV.

[14]Each secondary cloud will estimate the cost if it joins a coalition. The detailed modelling of coalition cost is out of scope of this research.

TABLE III: NE, coalition value and least pessimistic core (the outcome highlighted in yellow).

Coalition structure	NE request	Coalition value	Least ε	Payoff division
$\{1,2,3\}$	43.1	421.9	0	127.3, 119.2, 175.4
$\{1,2\},\{3\}$	23.1, 29.3	241.9, 181.2	1.1611	123.8, 118.1, 181.2
$\{1,3\},\{2\}$	47.6, 1.2	296.7, 119.3	5.90571	127.2, 119.3, 169.5
$\{1\},\{2,3\}$	29.4, 23.3	130.1, 292.7	0.941181	130.1, 118.2, 174.5
$\{1\},\{2\},\{3\}$	28.5, 0, 28.0	125.0, 119.2, 175.4	2.3508	125.0, 119.2, 175.4

TABLE IV: Pessimistic value

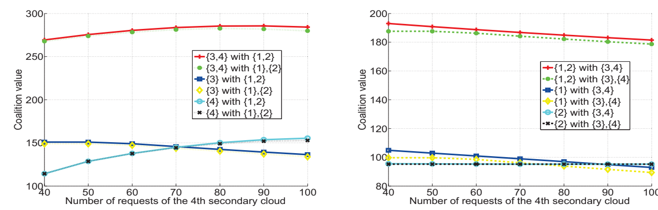
Coalition	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
Pessimistic value	125.0	119.2	175.4	241.9	296.7	292.7	421.9

Last, for each coalition structure, we call the linear program in Eqs.(11)-(14) to compute the least ε and the corresponding payoff of each secondary cloud, as is shown in the last two columns of Table III. We can see that only the least ε of the grand coalition is 0, which means that only the grand coalition and its corresponding payoff division is in the least pessimistic core (highlighted in yellow). Take another coalition structure $\{1\},\{2,3\}$ as an example, the payoff of secondary cloud 2 is 118.2, while the pessimistic value of coalition $\{2\}$ is $119.2 > 118.2$. Therefore, secondary cloud 2 has an incentive to deviate from the current outcome and form a singleton coalition. Interestingly, although secondary cloud 1 gets the highest payoff in this coalition structure, it is not stable due to secondary clouds 2's deviation.

B. Superadditivity and Externality

In this subsection, we show the superadditivity and externality of the CFG-e. We consider 4 secondary clouds and 1 primary cloud. Similarly, since we do not intend to evaluate the influence of the parameter values in this subsection, we assign arbitrary values for the parameters in this subsection. The primary cloud's capacity is set as 50. The penalty rates of the primary and secondary clouds are set as in the above subsection. The capacity, request and price vectors of the first three secondary clouds are also set as in the above subsection. The unit coalition cost c^{co} is set as 0 so that cooperation incurs no coalition cost. The capacity of secondary cloud 4 is set as 70. For different number of requests of secondary cloud 4, we evaluate 4 coalition structures: 1)$\{1,2\},\{3,4\}$; 2)$\{1,2\},\{3\},\{4\}$; 3)$\{1\},\{2\},\{3,4\}$; 4)$\{1\},\{2\},\{3\},\{4\}$.

In Fig. 5a, we show the values of coalitions $\{3,4\},\{3\}$ and $\{4\}$ when secondary clouds 1 and 2 cooperate or not. It shows that for all the three coalitions, the coalition value is always larger when secondary clouds 1 and 2 cooperate, which is consistent with the positive externality. Furthermore, for a certain cooperation status of secondary clouds 1 and 2, the value of coalition $\{3,4\}$ is always larger than the sum of values of coalitions $\{3\}$ and $\{4\}$, which implies supperadditivity. The same holds for Fig. 5b. In addition, we can see that in Fig. 5a, the values of coalitions $\{4\}$ and $\{3,4\}$ increase with the number of requests of secondary cloud 4. While this observation is straightforward, we also notice that in both Figs. 5a and 5b, the values of all the other coalitions which do not contain secondary cloud 4 decrease. This is because when the number of requests in secondary cloud 4 increases, it tends to outsource more requests to the primary cloud, which leads to congestion in the primary cloud and further deteriorates the task performance of other coalitions. This phenomenon indicates that the externality is caused by the resource contention in the primary cloud.



(a) Secondary clouds 3 & 4 (b) Secondary clouds 1 & 2

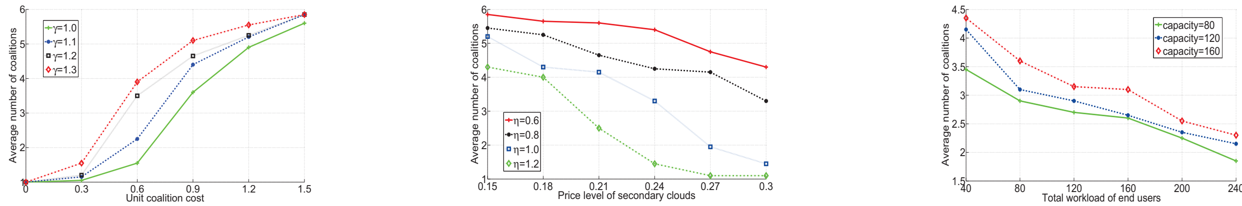
Fig. 5: Superadditivity & externality

C. Coalition Formation Results under Different Cloud Network Conditions

This subsection investigates several key factors that determine the coalition formation results. To approach the parameter values in real scenarios, in this subsection, we follow the pricing schema of Amazon EC2 [41], and assume that the clouds in the VHCF system use the t2.large VM in a Linux/UNIX operating system, with a per hour price of 0.104 US dollars (i.e., for all primary clouds $i \in \mathcal{I}$, $p_i^p = 0.104$). We consider 2 primary clouds and 6 secondary clouds. Each secondary cloud j's capacity is randomized within $[60, 100]$. Its request is randomized within $[c_i^s - 20, c_i^p + 60]$. Its price is randomized within $[p^s - 0.05, p^s + 0.05]$, where p^s indicates the overall price level of the secondary clouds and is randomized within $[0.15, 0.3]$.The price level is 1.5 to 3 times as much as the price of the primary cloud. Considering that we have not counted other costs such as labor, advertisement, and taxes, this is a reasonable range according to the definition of a "healthy operating profit margin" by Investopedia [42]. Each primary cloud i's capacity is randomized within $[40, 80]$. The penalty rate β^p of the primary clouds is randomized within $[0.08, 0.12]$. The penalty rate of the secondary clouds is set as $\beta^s = \eta p^s$, where η denotes the ratio of penalty against the price level and is randomized within $[0.8, 1.2]$ — by $\eta = 1$, it means that if the processing time is twice as much as the pre-contracted deadline, the penalty equals the price (which means the user does not need to pay for the request). The unit coalition cost c^{co} and the coalition size factor γ are randomized within $[0, 1.2]$ and $[1, 1.3]$, respectively.

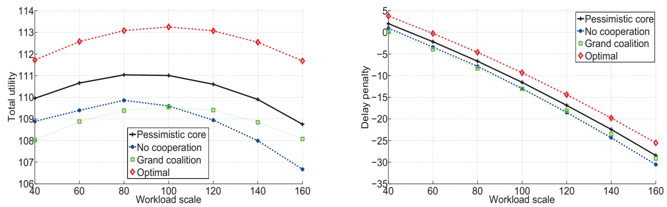
In this set of experiments, our strategy is to step by step vary the parameter value under evaluation, and randomize the values of other parameters over the above specified range. For each set of results in Fig. 6, we take the average over 30 trials.

We first evaluate the coalition formation results under different coalition cost. We set the unit cost from 0 to 1.5 with a step of 0.3, and observe coalition formation with different



(a) Different coalition cost (b) Different price and penalty rates (c) Different demand and supply

Fig. 6: Coalition formation results under different cloud network conditions



(a) Total utility (b) Total penalty

Fig. 7: Total utility and total penalty of the secondary clouds

coalition size factor γ. Fig. 6a shows the average number of coalitions formed under different coalition cost, where smaller number of coalitions means that the secondary clouds tend to cooperate. The figure indicates that when the unit coalition cost and the coalition size factor γ increase, the number of coalitions also increases. When coalition cost approaches 1.5, cooperation is too costly to form. Thus all secondary clouds tend to stay in singleton coalitions.

We then evaluate the coalition formation under different price and penalty rates of the secondary clouds. We set price level p^s of the secondary clouds from 0.15 to 0.3 with a step of 0.03, and evaluate the coalition formation with different secondary clouds' penalty rate η. As is shown in Fig. 6b, for a fixed η, the increase of price will promote the cooperation of the secondary clouds. This is because when the price of the secondary clouds increases, the coalition cost is less comparable with the profit. Therefore, cooperation will generate more benefit. Meanwhile, for a fixed price, when penalty rate increases, the secondary clouds tend to cooperate. As is indicated in the proof of Lemma 2, cooperation can always lead to no worse delay. Therefore, secondary clouds tend to cooperate to reduce the high delay penalty.

To investigate how the demand and supply relation influences coalition formation, we vary the request scale and capacity of the primary clouds. We define the request scale as the total number of requests received by the secondary clouds minus the total capacity of the secondary clouds. We set it from 40 to 240 with a step of 40, and evaluate its effect on coalition formation with different capacity scale of the primary clouds. Fig. 6c shows that when the secondary clouds have smaller capacity or there are more requests from the end users, the secondary clouds tend to cooperate. This is because when the total workload increases or the capacity of the primary clouds decreases, the delay of processing time will be larger. Therefore, the secondary clouds cooperate to share resources and reduce delay.

D. Total Utility Comparison

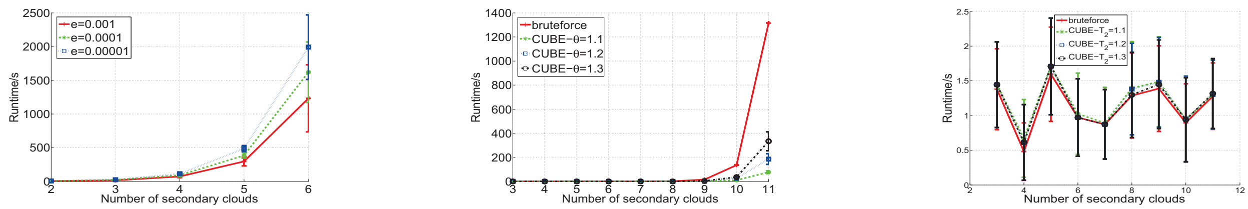
In this subsection, we answer the last question in Section I: is it beneficial for the secondary clouds to use a VHCF network? We compare the secondary clouds' total utility and total delay penalty with different request scale for the following 4 cases: 1) VHCF network; 2) no cooperation (the case in [2]); 3) grand coalition (the common practice of horizontal cloud federation); 4) optimal utility (workload factoring and coalition formation are optimized by a mediator). We consider 2 primary clouds and 5 secondary clouds. Except the capacity and request scales, all the parameter values (please refer to Table II) are set as in the above subsection. We fix the capacity scale of the primary clouds as 100, and investigate different request scales from 40 to 160.

As shown in Figs. 7a and 7b, the least pessimistic core generates larger total utility than that of no-cooperation and grand coalition. The total utility is higher when the workload scale is closer to the capacity scale of the primary clouds because when the workload is low, the gross revenue is low (although there is a bonus of finishing the requests in advance), but when the workload is too high, the penalty delay is too large. By using the VHCF network, the secondary clouds (using the t2.large VM from Amazon EC2 [41], with an operating scale of 40 to 160 VMs) can averagely decrease the delay penalty by 10.6% and 11.0%, compared with no-cooperation and grand coalition, respectively. The strategic decision making of the secondary clouds naturally leads to a sub-optimal total utility. Unfortunately, the pessimistic core bears an inevitable loss compared with the optimal case, since each cloud provider is self-interested.

E. Compare Different Algorithms

Last, we compare the runtime and optimality of different algorithms. In this subsection, all algorithms are implemented using Java, all linear programs are solved with CPLEX (v12.6) and all non-linear programs are solved with KNITRO (v9.0.0). All computations are performed on a 64-bit machine with 16 GB RAM and a quad-core Intel i7-4770 3.4 GHz processor. We consider 2 primary clouds and vary the number of secondary clouds. All the parameters are randomized as in Subsection VI-C. All the results are averaged with 20 trials.

Fig. 8a shows the runtime of Brute-Force with CLONE under different convergence threshold e, from 0.001 to 0.00001. Intuitively, smaller e leads to longer runtime, which increases rapidly with the number of secondary clouds. Fig. 8b shows the runtime of Brute-Force and CUBE (with different pruning thresholds θ) with CLOSE. We can see that comparing with Fig. 8a, the runtime of Brute-Force and CUBE with CLOSE is significantly reduced. When the total number of secondary clouds is smaller than 8, both of the two CLOSE-based algorithms take less than 1 second to finish. Furthermore, by



(a) Brute-Force with CLONE (b) Brute-Force and CUBE with CLOSE (c) Least core values of different algorithms.

Fig. 8: Runtime and least core values of different algorithms. A 90% confidence interval is shown.

implementing CUBE, the runtime has significantly reduced comparing with Brute-Force, while maintaining a close-to-optimal least ε value (Fig. 8c). With smaller θ, it takes less time because more nodes are pruned during the searching. By setting different pruning threshold, we can tradeoff between runtime and optimality. Using these algorithms, we can solve practical scale problems.

VII. Conclusion

The integration of vertical and horizontal cloud federation is a promising approach to improve a cloud's service quality. Notwithstanding the merits, the economic aspects of such a VHCF network is rarely investigated, especially when the secondary clouds are self-interested. With this work, we make the first attempt to reveal the interrelated workload factoring and coalition formation among secondary clouds, and answer the following intrinsic questions: 1) What is the most stable coalition structure and payoff division; 2) What are the optimal (equilibrium) workload factoring strategies of different coalitions; 3) Is it beneficial for the secondary clouds to join such a VHCF network? The experimental results show that interestingly, the two common practices in the real world - no-cooperation and the grand coalition - are sometimes not the most stable cooperation pattern. Instead, the most stable coalition structure usually lies between these two extreme cases. Furthermore, the experimental results show that by implementing the VHCF network, the service delay penalty of the secondary clouds decreases by around 11% compared with the two common practices. The runtime and optimality evaluation of the algorithms shows that our proposed algorithms can be applied to real world sized problems, and assist secondary clouds to choose their partners and decide the optimal workload factoring strategies. Our framework can be extended to a variety of network-related domains such as cellular networks, cognitive radio networks and smart grid, where cooperative and competitive behaviors coexist.

Acknowledgment

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative; and is supported in part by Singapore MOE Tier 1 (RG18/13 and RG33/12) & MOE Tier 2 (MOE2014-T2-2-015 ARC4/15 and MOE2013-T2-2-070 ARC16/14).

References

[1] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Algorithms and architectures for parallel processing*, pages 13–31. Springer, 2010.

[2] Amir Nahir, Ariel Orda, and Danny Raz. Workload factoring with the cloud: A game-theoretic perspective. In *INFOCOM*, pages 2566–2570, 2012.

[3] David Villegas, Norman Bobroff, Ivan Rodero, Javier Delgado, Yanbin Liu, Aditya Devarakonda, Liana Fong, S Masoud Sadjadi, and Manish Parashar. Cloud federation in a layered service model. *Journal of Computer and System Sciences*, 78(5):1330–1344, 2012.

[4] Bill Steele. Aristotle: A federated cloud for academic research. http://www.news.cornell.edu/stories/2015/11/aristotle-federated-cloud-academic-research, 2015.

[5] Eubrazil cloud connect. http://www.eubrazilcloudconnect.eu/.

[6] Michael P Papazoglou. Cloud blueprints for integrating and managing cloud federations. In *Software Service and Application Engineering*, pages 102–119. Springer, 2012.

[7] Francesco Tusa, Antonio Celesti, Maurizio Paone, Massimo Villari, and Antonio Puliafito. How clever-based clouds conceive horizontal and vertical federations. In *ISCC*, pages 167–172, 2011.

[8] Di Niu and Baochun Li. An efficient distributed algorithm for resource allocation in large-scale coupled systems. In *INFOCOM*, pages 1501–1509, 2013.

[9] Wei Wang, Baochun Li, and Ben Liang. Dominant resource fairness in cloud computing systems with heterogeneous servers. In *INFOCOM*, pages 583–591, 2014.

[10] Hongxing Li, Chuan Wu, Zongpeng Li, and Francis CM Lau. Profit-maximizing virtual machine trading in a federation of selfish clouds. In *INFOCOM*, pages 25–29, 2013.

[11] Liang Zhang, Weijie Wu, and Dan Wang. Time dependent pricing in wireless data networks: Flat-rate vs. usage-based schemes. In *INFOCOM*, pages 700–708, 2014.

[12] Ruben Van den Bossche, Kurt Vanmechelen, and Jan Broeckhove. Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In *CLOUD*, pages 228–235, 2010.

[13] Wei-Jen Wang, Yue-Shan Chang, Win-Tsung Lo, and Yi-Kang Lee. Adaptive scheduling for parallel tasks with qos satisfaction for hybrid cloud environments. *The Journal of Supercomputing*, 66(2):783–811, 2013.

[14] Yuan Feng, Baochun Li, and Bo Li. Price competition in an oligopoly market with multiple iaas cloud providers. *IEEE Transactions on Computers*, 63(1):59–73, 2014.

[15] Francesco Palmieri, Luigi Buonanno, Salvatore Venticinque, Rocco Aversa, and Beniamino Di Martino. A distributed scheduling framework based on selfish autonomous agents for federated cloud environments. *Future Generation Computer Systems*, 29(6):1461–1472, 2013.

[16] Dusit Niyato, Athanasios V Vasilakos, and Zhu Kun. Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach. In *CCGrid*, pages 215–224, 2011.

[17] Lena Mashayekhy and Daniel Grosu. A coalitional game-based mechanism for forming cloud federations. In *UCC*, pages 223–227, 2012.

[18] Marco Guazzone, Cosimo Anglano, and Matteo Sereno. A game-theoretic approach to coalition formation in green cloud federations. In *CCGrid*, pages 618–625, 2014.

[19] Sudarshan Guruacharya, Dusit Niyato, Mehdi Bennis, and Dong In Kim. Dynamic coalition formation for network mimo in small cell networks. *IEEE Transactions on Wireless Communications*, 12(10):5360–5372, 2013.

[20] Tianyu Wang, Lingyang Song, Zhu Han, and Walid Saad. Distributed cooperative sensing in cognitive radio networks: An overlapping coalition formation approach. *IEEE Transactions on Communications*, 62(9):3144–3160, 2014.

[21] Yong Xiao, Dusit Niyato, Zhu Han, and Kwang-Cheng Chen. Secondary users entering the pool: A joint optimization framework for spectrum pooling. *IEEE Journal on Selected Areas in Communications*, 32(3):572–588, 2014.

[22] Woongsup Lee, Lin Xiang, Robert Schober, and Vincent WS Wong. Direct electricity trading in smart grid: a coalitional game analysis. *IEEE Journal on Selected Areas in Communications*, 32(7):1398–1411, 2014.

[23] Richard Southwell, Xu Chen, and Jianwei Huang. Quality of service games for spectrum sharing. *IEEE Journal on Selected Areas in Communications*, 32(3):589–600, 2014.

[24] Wei Zhong, Yixin Fang, Shi Jin, Kai-Kit Wong, Sheng Zhong, and Zuping Qian. Joint resource allocation for device-to-device communications underlaying uplink mimo cellular networks. *IEEE Journal on Selected Areas in Communications*, 33(1):41–54, 2015.

[25] Mina Fahimi and Abdorasoul Ghasemi. Joint spectrum load balancing and handoff management in cognitive radio networks: a non-cooperative game approach. *Wireless Networks*, pages 1–20, 2015.

[26] Eddy Caron, Frederic Desprez, and Adrian Muresan. Forecasting for grid and cloud computing on-demand resources based on pattern matching. In *CloudCom*, pages 456–463, 2010.

[27] Amazon ec2 service level agreement. https://aws.amazon.com/ec2/sla/, 2016.

[28] Microsoft Azure. Sla for virtual machines. https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/v1_2/, 2016.

[29] Pankesh Patel, Ajith H Ranabahu, and Amit P Sheth. Service level agreement in cloud computing. 2009.

[30] L. Kóczy. A recursive core for partition function form games. *Theory and Decision*, 63(1):41–51, 2007.

[31] Lloyd S Shapley. A value for n-person games. Technical report, DTIC Document, 1952.

[32] Georgios Chalkiadakis, Evangelos Markakis, and Nicholas R Jennings. Coalitional stability in structured environments. In *AAMAS*, pages 779–786, 2012.

[33] Thanh H Nguyen, Amulya Yadav, Bo An, Milind Tambe, and Craig Boutilier. Regret-based optimization and preference elicitation for stackelberg security games with uncertainty. In *AAAI*, pages 756–762, 2014.

[34] Chee Shin Yeo and Rajkumar Buyya. Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. In *2005 IEEE International Conference on Cluster Computing*, pages 1–10. IEEE, 2005.

[35] Omer F Rana, Martijn Warnier, Thomas B Quillinan, Frances Brazier, and Dana Cojocarasu. Managing violations in service level agreements. In *Grid Middleware and Services*, pages 349–358. Springer, 2008.

[36] Linlin Wu, Saurabh Kumar Garg, and Rajkumar Buyya. Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. In *CCGrid*, pages 195–204, 2011.

[37] Shizuo Kakutani et al. *A generalization of Brouwer's fixed point theorem*. Duke University Press, 1941.

[38] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1):209–238, 1999.

[39] Ning Chen, Pinyan Lu, and Hongyang Zhang. Computing the nucleolus of matching, cover and clique games. In *AAAI*, 2012.

[40] Andreas S Schulz and Nelson A Uhan. Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optimization*, 10(2):163–180, 2013.

[41] Amazon. Amazon ec2 pricing. https://aws.amazon.com/ec2/pricing/, 2016.

[42] Investopedia. What is considered a healthy operating profit margin? http://www.investopedia.com/ask/answers/041415/what-considered-healthy-operating-profit-margin.asp, 2015.

Haipeng Chen is a Ph.D. student in the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore. He received the B.S. in School of Physics, University of Science and Technology of China, Hefei, China, in 2013. His research interests are in the area of game theory, multi-agent systems, artificial intelligence, machine learning and cloud computing.

Bo An is a Nanyang Assistant Professor at the School of Computer Science and Engineering of the Nanyang Technological University. He received the Ph.D degree in Computer Science from the University of Massachusetts, Amherst and was a Postdoctoral Researcher at the University of Southern California. His research interests include artificial intelligence, multi-agent systems, game theory, and optimization. He has published over 50 referred papers at AAMAS, IJCAI, AAAI, ICAPS, KDD, JAAMAS and IEEE Transactions. He was the recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, an Operational Excellence Award from the Commander, First Coast Guard District of the United States, the Best Innovative Application Paper Award at AAMAS-12, the 2012 INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice, and the Innovative Application Award at IAAI-16. He is a member of the editorial board of JAIR and the Associate Editor of JAAMAS. He was elected to the board of directors of IFAAMAS.

Dusit Niyato (M'09-SM'15-F'17) is currently an Associate Professor in the School of Computer Science and Engineering, at Nanyang Technological University, Singapore. He received B.Eng. from King Mongkuts Institute of Technology Ladkrabang (KMITL), Thailand in 1999 and Ph.D. in Electrical and Computer Engineering from the University of Manitoba, Canada in 2008. His research interests are in the area of energy harvesting for wireless communication, Internet of Things (IoT) and sensor networks.

Chuanyan Miao is a Professor in the School of Computer Science and Engineering at Nanyang Technological University. She is the director of the NTU-UBC Joint Research Centre of Excellence in Active Living for the Elderly (LILY). Prior to joining NTU, she was an Instructor and Postdoctoral Fellow at the School of Computing, Simon Fraser University, Canada. Her research focus is on studying the cognitive and social characteristics of intelligent agents in multi-agent and distributed AI/CI systems, such as trust, emotions, motivated learning, ecological and organizational behavior. She has made significant contributions in the integration of the above research into emerging technologies such as interactive digital media (e.g., virtual world, social networks, and massively multi-player game), cloud computing, mobile communication, and humanoid robots.

Yeng Chai Soh received the B.Eng. (Hons. I) degree in electrical and electronic engineering from the University of Canterbury, New Zealand, and the Ph.D. degree in electrical engineering from the University of Newcastle, Australia. He joined the Nanyang Technological University, Singapore, after his PhD study and is currently a professor in the School of Electrical and Electronic Engineering. Dr Soh has served as the Head of the Control and Instrumentation Division, the Associate Dean (Research and Graduate Studies) and the Associate Dean (Research) at the College of Engineering. Dr Soh has served as panel members of several national grants and scholarships evaluation and awards committees. Dr Soh's research interests have been in robust control and applications, robust estimation and filtering, optical signal processing, and energy efficient systems. He has published more than 260 refereed journal papers in these areas. His most recent research projects and activities are in sensor networks, sensor fusion, distributed control & optimization, and control & optimization of ACMV systems.