Received 11 December 2019; revised 17 January 2020; accepted 18 January 2020. Date of publication 27 January 2020; date of current version 18 February 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.2969646

A Game-Theoretic Analysis of Shared/Buy-in Computing Systems

ZHENPENG SHI [1], AZER BESTAVROS [2], ARIEL ORDA [3], AND DAVID STAROBINSKI [1]

[1]Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA

[2]Department of Computer Science, Boston University, MA 02215 USA

[3]Viterbi Faculty of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 3200003, Israel

CORRESPONDING AUTHOR: Z. SHI (e-mail: zpshi@bu.edu)

This work was supported in part by the U.S. National Science Foundation under Grant CNS-1717858 and Grant CNS-1908087, and in part by the U.S.–Israel Binational Science Foundation under Grant NSF-BSF-2016690.

ABSTRACT High performance computing clusters are increasingly operating under a shared/buy-in paradigm. Under this paradigm, users choose between two tiers of services: shared services and buy-in services. Shared services provide users with access to shared resources for free, while buy-in services allow users to purchase additional buy-in resources in order to shorten job completion time. An important feature of shared/buy-in computing systems consists of making unused buy-in resources available to all other users of the system. Such a feature has been shown to enhance the utilization of resources. Alongside, it creates strategic interactions among users, hence giving rise to a non-cooperative game at the system level. Specifically, each user is faced with the questions of whether to purchase buy-in resources, and if so, how much to pay for them. Under quite general conditions, we establish that a shared/buy-in computing game yields a unique Nash equilibrium, which can be computed in polynomial time. We provide an algorithm for this purpose, which can be implemented in a distributed manner. Moreover, by establishing a connection to the theory of aggregative games, we prove that the game converges to the Nash equilibrium through best response dynamics from any initial state. We justify the underlying game-theoretic assumptions of our model using real data from a computing cluster, and conduct numerical simulations to further explore convergence properties and the influence of system parameters on the Nash equilibrium. In particular, we point out potential unfairness and abuse issues and discuss solution venues.

INDEX TERMS Computing clusters, dynamics, equilibrium analysis, pricing.

I. INTRODUCTION

IN ORDER to achieve economy of scale, major research institutions are increasingly consolidating their IT services into High Performance Computing (HPC) clusters. HPC clusters make use of advanced parallel computing tools, such as Apache Hadoop [1], to join the computational powers of multiple computing nodes and provide a powerful computing environment. For example, the Boston University Shared Computing Cluster (BU SCC) is a heterogeneous HPC cluster that supports hundreds of research projects [2]. The size of the BU SCC has grown rapidly, from 189 computing nodes in 2013 to 835 nodes in 2019, to support the high demand for computing resources.

Many HPC clusters have adopted a *shared/buy-in computing* paradigm for their services. In particular, universities commonly provide free (shared) resources to staffs and students for research. Yet, in several domains (e.g., biology, medicine, and physics), researchers have additional computational needs. As a result, many universities implement a shared/buy-in paradigm, which allows researchers to buy-in additional resources. Examples of academic institutions which have adopted this paradigm include the HPC clusters at Boston University [2], Northeastern University [3], the University of Wisconsin-Madison [4], the University of Arizona [5], and the University of California, Berkeley [6].

A shared/buy-in computing system consists therefore of shared and buy-in computing resources. All users can make use of shared resources for free, typically on a fair-share allocation basis. In addition, users can elect to acquire buy-in computing nodes for their research projects. Buy-in nodes are operated under a *semi-exclusive* policy. Priority access is given to owners of those nodes, but excess idle capacity is made accessible to other users (both buy-in and shared users). In other words, the buy-in resources of a user are not fully exclusive to itself; rather, the advantage to the buyer is in terms of priority access to these resources. This policy has been shown to enhance the utilization of resources and lower the total amount of resources needed for operating the system [7].

Due to the semi-exclusive buy-in feature of shared/buy-in computing systems, users will interact with each other through their decisions. The strategic interaction exists in the sense that if a user decides to buy a large amount of resources, others have less motivation to buy since they can access more resources thanks to that user. It is reasonable to assume that users in such systems behave in a rational, selfish manner to optimize their own objectives, thus giving rise to a non-cooperative game among users, which is the main focus of this study. We assume that each user needs to complete a job using the resources of a shared/buy-in computing system. A user needs to decide whether to purchase its own buy-in nodes, and how much it is willing to pay in that case. We propose a non-cooperative game-theoretic model to analyze user behavior in such systems.

The first step is to find if such systems admit a Nash equilibrium. We establish that the game considered in this study does admit a unique Nash equilibrium, and, furthermore, we show that it can be computed in polynomial time.

Next, we investigate the dynamics of the game. By establishing a connection to the theory of aggregative games with strategic substitutes [8], [9], we manage to prove convergence of best response dynamics from any initial state in a general game with N players. We also show that each player can compute its best response in a distributed manner. This result implies that the unique Nash equilibrium not only exists, but is also likely to be reached. Through numerical simulations based on actual data from the BU SCC, we confirm our theoretical model and implied results and explore additional properties of the game. Finally, we discuss how the system parameters influence the game's Nash equilibrium, and indicate the existence of opportunities for users to abuse resources in shared/buy-in computing systems along with guidelines for addressing such problems.

The main contributions of this paper can be summarized as follows:

- We formulate a new game-theoretic model to analyze shared/buy-in computing systems.
- We establish the existence and uniqueness of the game's Nash equilibrium.
- We design and validate an efficient, polynomial-time algorithm for calculating the Nash equilibrium of the game in the general, N-player case.
- We establish convergence properties of best response dynamics in the general, N-player case from an arbitrary initial state.
- We investigate other convergence properties, such as convergence speed, through numerical simulations.
- We confirm our modeling assumptions regarding the rational behavior of users, using actual data from the BU SCC.
- We investigate the influence of various parameters on the Nash equilibrium and point out potential problems in the investigated system as well as solution venues.
- Our results provide the following insights: (i) the Nash equilibrium can be computed faster using best response dynamics; (ii) increasing the amount of buy-in resources that a user gets per currency unit may lead some users to pay less and other users to pay more; (iii) users with small workloads generally benefit more from the system than users with larger workloads, which may lead to the emergence of free-riders.

The rest of this paper is organized as follows. After reviewing related work in Section II, we formalize our model for shared/buy-in computing game in Section III. Next, we establish the existence and uniqueness of the Nash equilibrium in Section IV. Convergence of best response dynamics is studied in Section V. Then, in Section VI, we present numerical results to illustrate, confirm and expand our theoretical results. Conclusions are presented in Section VII.

II. RELATED WORK

Much work has been devoted to characterize the workload of computing clusters, e.g., [10], [11]. While most of the work does not involve shared/buy-in computing systems, a study of workload characterization of the BU SCC is particularly related to our work [7]. In that study, the typical behavior and performance of a shared/buy-in computing system is characterized using data traces from the BU SCC. It is shown that, as expected, the semi-exclusive policy increases the utilization of buy-in resources. Moreover, that study characterizes several statistical patterns of the SCC. We leverage these statistical characterizations in our simulations.

Game-theoretic approaches have been adopted widely in areas that involve users' strategic interaction, such as internetworking [12], wireless networks [13], shared spectrum access [14], cybersecurity [15], distributed computing [16], cloud co-location services [17], and advanced reservations [18]. The main goal in those studies has been to find the Nash equilibrium of the respective games.

Pricing of cloud services is yet another interesting area where game theory has proved useful. An overview of pricing models in cloud networking with their applications for resource management is presented in [19]. Specifically, many pricing models are based on a game-theoretic perspective in order to study the strategic interaction between cloud providers or cloud users. For example, two pricing schemes for cloud services are discussed in [20], namely fixed and spot-market-based pricing. It is shown that, when both pricing schemes are available, users employ a waiting cost threshold to determine in which scheme to participate. As a result, the provider's expected revenue is lower than when adopting only fixed pricing. In terms of shared/buy-in computing systems, we investigate prices paid by users, and determine that there exists a workload threshold that dictates whether a user will elect to purchase buy-in resources.

Another study related to our work is [21], in which a *workload factoring* game is investigated. In that model, users can split their work between shared resources and private resources. While shared resources cannot provide guaranteed performance due to other players' influence, they are more powerful than private resources. Each user should choose a strategy on how to split its workload between the private and shared resources so as to minimize its job's completion time. In a shared/buy-in computing system users split their work between shared resources and buy-in resources, which can also be seen as an instance of workload factoring; however, in our framework a *price* is paid for the buy-in resources, hence the primary consideration is how to minimize the related cost. The work of [21] does not incorporate any such economic considerations. Another difference is that our model is a continuous one, while the model of [21] bears a discontinuity property. Hence, different analytical approaches are needed to investigate the game's equilibrium.

The dynamic behavior of the game is another important property. If a game has a Nash equilibrium but cannot converge to it in reasonable time, or cannot return to its Nash equilibrium once deviating from it due to minor disturbance, then the practical relevance of the Nash equilibrium may be quite limited. In our study, we consider the dynamic behavior of the game model both theoretically and numerically, and show that the game will not only converge to its Nash equilibrium, but also do so quite fast. It is also worth noting that our theoretical results are confirmed through data from a real-world system.

In the literature, some classes of non-cooperative games have attracted particular attention due to their special convergence properties; one such class is that of supermodular games [22]. These games can be used to characterize strategic complements, that is, when a player takes a higher action in a game, other players tend to do the same [23]. Supermodular games have many interesting properties, such as existence of a pure strategy Nash equilibrium, and they often arise in networking contexts, for example, power control schemes for wireless networks [24].

In contrast to supermodular games, submodular games have received less attention. In these games, each player maximizes a submodular function, and strategic substitutes prevail instead of strategic complements. In general, much less is known about this class of games. There have been some attempts to generalize the supermodular model so that it allows supermodular and submodular games to coexist, leading to the notion of S-modularity [25]. Indeed, with S-modularity, several interesting results have been established in the area of power control [26]. However, S-modularity still requires supermodularity at some level. In our context, namely of shared/buy-in computing systems, supermodularity can only be achieved in the 2-player case, after some transformation, but not in the general N-player case.

Aggregative games represent another important class of games closely related to our framework. In an aggregative game, a player's payoff depends on the "aggregate" of all players' strategies rather than the opponents' individual strategies [8]. One example is the Cournot oligopoly model, where a player's payoff is a function of the sum of all player's supply quantities. In [9], it is shown that, in an aggregative game with strategic complements or substitutes, a pure strategy Nash equilibrium must exist (the proof is not constructive, however, and the equilibrium may not be unique). Under certain conditions, it is proven that an aggregative game converges to a Nash equilibrium [9], [27]. Quasi-aggregative games are proposed in [28], which allows the "aggregate" to take different forms other than a simple sum. The work in [28] also extends relevant results about Nash equilibrium and convergence using the notion of best-response potential games [29]. In our paper, we prove that our model is in fact an aggregative game with strategic substitutes, and thus it admits established properties of such games.

The work in [30] analyzes a voluntary contribution model of pure public good provision. In this aggregative game, each player can make use of public good, and needs to decide how much to contribute to it. This public good model is quite general and has a wide range of applications, such as in environmental problems [31]. While computing resources are a type of public good, our model differs in that it captures unique features of shared/buy-in computing systems. As a result, we establish the existence and uniqueness of the Nash equilibrium in a constructive manner, while the model of [30] does so in a non-constructive manner. Moreover, the model of [30] focuses on the comparative static properties of the equilibrium as players' incomes and unit cost of contribution change, while our model focuses on aspects that are more specific to shared/buy-in computing systems, such as how the strategy of a user is influenced by its workload, other players' workloads, and various system parameters (see Table 1).

III. SYSTEM MODEL

In this section, we introduce our system model and parameters. We illustrate practical settings of the parameters in Section VI. The notations are summarized in Table 1.

We consider a strategic game in the form of $\langle \Phi, \{P_i\}_{i\in\Phi,}, \{U_i\}_{i\in\Phi} \rangle$, where Φ is the finite set of players, P_i is the non-empty strategy set of player i to which its strategy p_i belongs, and $U_i : P \to \mathbb{R}$ is the payoff of player i given a strategy profile of all players from the joint set $P = \prod_{i\in\Phi} P_i$.

In our setting of shared/buy-in computing (SBC) games, there are N players in total, and each player is a user that has

TABLE 1. Notation summary.

Notation	Description
Φ	Set of players (users).
N	Number of players.
i, j, ℓ, m	Index of players.
p_i	Strategy (price paid for buy-in resources) of player i.
P_i	Strategy set of player i.
$\mathbf{p}$	Strategy profile of all players.
$U_i(\mathbf{p})$	Payoff of player i given $\mathbf{p}$.
$C_i(\mathbf{p})$	Cost of player i given $\mathbf{p}$.
$T_i(\mathbf{p})$	Job completion time of player i given $\mathbf{p}$.
ω_i	Average workload of player i.
μ	Computing rate provided to each player with shared resources.
k_b	Coefficient of proportionality between the price paid by player i and the corresponding computing rate it gets.
k_s	Coefficient of proportionality between the price paid by player i and the corresponding computing rate it provides to another player.
α	Coefficient that reflects the player's sensitivity to price versus job completion time.

a certain amount of work (workload) to complete (e.g., in units of CPU-hours). Note that in real-world shared/buy-in computing systems, a "user" may not necessarily be a single individual. For instance, the SCC at Boston University is managed by "projects". In that case, each project corresponds to a player.

Each player i has an average workload ω_i and a strategy $p_i \geq 0$, which corresponds to the price that it pays for buy-in resources. The average workload of players is estimated based on the aggregate workload measured over a long time period. Thus, when we validate our results using BU SCC data in Section VI, the average workload is estimated based on the aggregate workload measured over a period of seven months. Note that by using p_i to denote the price paid by player i, we capture the player's decision on two levels: (a) it will pay $p_i > 0$ if it decides to buy resources or $p_i = 0$ if not, and (b) how much to buy is reflected by the value of p_i. Let $\mathbf{p} = [p_1, p_2, \ldots, p_N]^\intercal$ denote the strategy profile of all players. Even though prices are discrete in practice, we make here the common assumption that they assume continuous values. In the rest of this paper, we assume without loss of generality that players are labelled such that

$$\omega_1 \geq \omega_2 \geq \cdots \geq \omega_N.$$

Note that one can always relabel the players to satisfy this property.

We assume that the SBC game is "fair" in the sense that each player gets the same computing rate μ out of the shared resources. For instance, in the BU SCC, it is the case that each user gets by default the same allocation of shared resources.

When a player buys in resources at a price p_i, it gets an additional computing rate of $k_b p_i$. Alongside, it provides each of the other players with an additional computing rate $k_s p_i$, since idle buy-in resources are available to other players. In other words, k_b and k_s are coefficients that convert the price paid by player i into the corresponding computing rate provided to player i itself and another player, respectively. We make the reasonable assumption that $k_b > (N-1)k_s$, that is, for a price p_i, player i gets a computing rate $k_b p_i$ that is larger than the aggregate computing rate $(N-1)k_s p_i$ provided to all other players. For instance, in the BU SCC, the aggregate workload of buy-in resources utilized by their owners is about twice larger than the aggregate workload of buy-in resources utilized by other users [7].

As a result, the total computing rate of a player i is $\mu + k_b p_i + \sum_{j \neq i} k_s p_j$ and its expected job completion time for a given workload ω_i is

$$T_i(\mathbf{p}) = \frac{\omega_i}{\mu + k_b p_i + \sum_{j \neq i} k_s p_j}, \tag{1}$$

where $i, j \in \{1, 2, \ldots, N\}$. Note that different players do not have to start their jobs at the same time, because we only consider average behavior (i.e., our model suggests that a player always provides an additional computing rate $k_s p_i$ to each of the other players.)

Each player contemplates two types of costs, namely, the job completion time and the price it pays for buy-in resources. Specifically, the total cost of a player i is defined as

$$\begin{aligned} C_i(\mathbf{p}) &= T_i(\mathbf{p}) + \alpha p_i \\ &= \frac{\omega_i}{\mu + k_b p_i + \sum_{j \neq i} k_s p_j} + \alpha p_i \end{aligned} \tag{2}$$

where $i, j \in \{1, 2, \ldots, N\}$, and α is a coefficient that reflects the user's sensitivity to price versus job completion time. Thus, the objective of each player i is to choose a proper price p_i in order to minimize its cost $C_i(\mathbf{p})$.

Although we assume that a player's objective is to minimize its cost, we can actually express also the payoff of player i as the difference between its gross payoff $\Gamma(\omega_i)$ by completing its workload ω_i and its cost:

$$\begin{aligned} U_i(\mathbf{p}) &= \Gamma(\omega_i) - C_i(\mathbf{p}) \\ &= \Gamma(\omega_i) - \frac{\omega_i}{\mu + k_b p_i + \sum_{j \neq i} k_s p_j} - \alpha p_i, \end{aligned}$$

where $i, j \in \{1, 2, \ldots, N\}$. Note that the gross payoff $\Gamma(\omega_i)$ is independent of the prices paid by players, so we get

$$\text{argmax}_{p_i \in P_i} U_i(\mathbf{p}) = \text{argmin}_{p_i \in P_i} C_i(\mathbf{p}),$$

which implies that maximizing the payoff is equivalent to minimizing the cost.

Remark 1: For simplicity, in this paper, we assume that the coefficient α is the same for all users. However, one can easily relax this assumption. If each player i has a different α_i, the cost becomes

$$\begin{aligned} C_i^*(\mathbf{p}) &= \frac{\omega_i}{\mu + k_b p_i + \sum_{j \neq i} k_s p_j} + \alpha_i p_i \\ &= \frac{\alpha_i}{\alpha}\left(\frac{\omega_i \alpha / \alpha_i}{\mu + k_b p_i + \sum_{j \neq i} k_s p_j} + \alpha p_i\right) \\ &= \frac{\alpha_i}{\alpha}\left(\frac{\omega_i^*}{\mu + k_b p_i + \sum_{j \neq i} k_s p_j} + \alpha p_i\right). \end{aligned}$$

Thus, we can get a new cost function for player i by replacing ω_i with ω_i^* and multiplying the entire expression by α_i/α. We note that this multiplicative constant has no bearing on the optimal strategy that minimizes the cost function. Hence, we can still get most of the results derived in Sections IV and V with small modifications, including the existence and uniqueness of the Nash equilibrium, an algorithm to compute the Nash equilibrium, and convergence properties.

Remark 2: Note each player i gets priority access to its own buy-in resources; hence, $k_b p_i$ is close to the maximum computing rate provided by its buy-in resources. Since a user does not use its own buy-in resources all the time, it will provide each of the other users with an average computing rate $k_s p_i$.

We can assume here a constant k_s for all players for two reasons: (a) the payoff of a user depends only on its own strategy and the sum of the other players' strategies, and (b) the number of users N in a shared/buy-in computing system is typically large, namely in the order of hundreds or thousands.

In the following, we shall detail why a constant k_s is enough for calculating the players' approximate costs. For this, let assume that when each player i pays p_i, it provides each of the other players with an additional computing rate $k_{si}p_i$ instead of $k_s p_i$ and $k_b > (N-1)k_{si}$. Hence the actual cost of player i is

$$\begin{aligned} C_i^{**}(\mathbf{p}) &= \frac{\omega_i}{\mu + k_b p_i + \sum_{j\neq i} k_{sj} p_j} + \alpha p_i \\ &= \frac{\omega_i}{\mu + (k_b - k_{si})p_i + \sum_{j\in\Phi} k_{sj} p_j} + \alpha p_i. \end{aligned}$$

Next, let k_s be defined such that

$$k_s \sum_{j\in\Phi} p_j = \sum_{j\in\Phi} k_{sj} p_j. \tag{3}$$

Since the number of players N is typically large, from the assumption that $k_b > (N-1)k_s$ and $k_b > (N-1)k_{si}$, we get $k_b \gg k_s$ and $k_b \gg k_{si}$, respectively. As a result, we have

$$(k_b - k_s)p_i \approx (k_b - k_{si})p_i. \tag{4}$$

Combining (3) and (4), we get

$$\begin{aligned} C_i(\mathbf{p}) &= \frac{\omega_i}{\mu + k_b p_i + \sum_{j\neq i} k_s p_j} + \alpha p_i \\ &= \frac{\omega_i}{\mu + (k_b - k_s)p_i + k_s \sum_{j\in\Phi} p_j} + \alpha p_i \\ &\approx \frac{\omega_i}{\mu + (k_b - k_{si})p_i + \sum_{j\in\Phi} k_{sj} p_j} + \alpha p_i \\ &= C_i^{**}(\mathbf{p}), \end{aligned}$$

which implies that, by assuming a constant k_s, the cost of each player is approximately the same as the actual cost.

IV. NASH EQUILIBRIUM ANALYSIS

In this section, we prove that an SBC game yields a unique (pure) Nash equilibrium. We also provide an efficient algorithm to compute the equilibrium. In the following, all the indices i, j, ℓ, m belong to the set $\{1, 2, \ldots, N\}$, where N is the number of players in the game.

First, we consider the player's *best response*, which is defined as the player's optimal strategy given all other players' strategies. The best response strategy p_i of player i is to minimize its cost (2) given $\{p_j \mid j \neq i\}$, namely:

$$p_i = \max\left(0, \sqrt{\frac{\omega_i}{\alpha k_b}} - \frac{k_s}{k_b}\sum_{j\neq i} p_j - \frac{\mu}{k_b}\right). \tag{5}$$

Note that the strategy of player i is continuous by assumption, and its best response is upper-bounded by $\sqrt{\frac{\omega_i}{\alpha k_b}} > 0$ and lower-bounded by 0, which implies that P_i is non-empty and compact.

Here, we implicitly assume that each player i knows the sum of prices paid by other players $\sum_{j\neq i} p_j$ in order to calculate its best response. This can be achieved in a *centralized* way, for example, if the service provider directly provides information of prices paid by all users. Yet it is also possible for players to find best responses in a *distributed* manner: the players have information of system parameters μ, k_b, k_s and have control over their own workloads ω_i, so each player can infer the sum of prices paid by other players by observing its own delay (1) (completion time of a new job). Besides, side-information that is accessible to users could also be employed to estimate the sum of prices paid by other players. For example, basic information about buy-in computing nodes in the BU SCC is publicly known, from which users can estimate the sum by themselves.

A *Nash equilibrium* is a point at which each player in the game is playing its best response to the other players' strategies, which implies that a player cannot lower its cost by unilaterally changing its strategy.

We proceed to establish the existence of the Nash equilibrium of the SBC game.

Lemma 1: An SBC game admits a Nash equilibrium.

Proof: First, for each player i, its best response (5) is upper-bounded by $\sqrt{\frac{\omega_i}{\alpha k_b}} > 0$ and lower-bounded by 0, so p_i can take any value from $[0, \sqrt{\frac{\omega_i}{\alpha k_b}}]$, which implies that P_i is compact and convex.

Next, observe that $\mu > 0$, $k_b > 0$, $k_s > 0$, and $p_i \geq 0$ for all i, so the payoff $U_i(\mathbf{p})$ of player i is continuous in $\mathbf{p}$. Moreover, taking the second derivative of payoff function $U_i(\mathbf{p})$ in terms of p_i, we obtain

$$\frac{\partial^2 U_i(\mathbf{p})}{\partial p_i^2} = -\frac{k_b^2 \omega_i}{\left(\mu + k_b p_i + \sum_{j\neq i} k_s p_j\right)^3} < 0,$$

which implies that $U_i(\mathbf{p})$ is strictly concave in p_i for fixed $\{p_j \mid j \neq i\}$. The lemma then follows from [32, Th. 1]. ■

Remark 3: In principle, it is possible to prove the uniqueness of the Nash Equilibrium by showing that the game satisfies the so-called "diagonally strictly concave property" [32]. Yet, given the generality of our model, we find it difficult to prove this property, even in the 2-player case.

Besides, in our analysis of the Nash equilibrium, we not only prove uniqueness, but also characterize properties of SBC games, which leads to an algorithm for computing the Nash equilibrium.

The next lemma shows that, at a Nash equilibrium, a player with a larger workload will pay no less than a player with a smaller workload.

Lemma 2: At a Nash equilibrium of an SBC game,

$$p_1 \geq p_2 \geq \cdots \geq p_N \geq 0.$$

Proof: Assume that Lemma 2 does not hold, that is, there exist i, j, such that, at a Nash equilibrium, $\omega_i \geq \omega_j$ and $p_i < p_j$. We distinguish between two cases: (A) $p_i = 0$, (B) $p_i > 0$, and analyze the best response strategies of players i and j in each of the two cases:

(A) $p_j > p_i = 0$. At a Nash equilibrium, the best response strategies of player i and player j must be according to (5)

$$\begin{cases} \frac{k_s}{k_b} p_j + \frac{k_s}{k_b} \sum_{m \neq i,j} p_m + \frac{\mu}{k_b} \geq \sqrt{\frac{\omega_i}{\alpha k_b}}, \\ p_j + \frac{k_s}{k_b} \sum_{m \neq i,j} p_m + \frac{\mu}{k_b} = \sqrt{\frac{\omega_j}{\alpha k_b}}. \end{cases} \tag{6}$$

(B) $p_j > p_i > 0$. At a Nash equilibrium, the best response strategies of player i and player j must be according to (5)

$$\begin{cases} p_i + \frac{k_s}{k_b} p_j + \frac{k_s}{k_b} \sum_{m \neq i,j} p_m + \frac{\mu}{k_b} = \sqrt{\frac{\omega_i}{\alpha k_b}}, \\ p_j + \frac{k_s}{k_b} p_i + \frac{k_s}{k_b} \sum_{m \neq i,j} p_m + \frac{\mu}{k_b} = \sqrt{\frac{\omega_j}{\alpha k_b}}. \end{cases} \tag{7}$$

However, in either case, (6) or (7) cannot hold if $\omega_i \geq \omega_j$ and $k_s/k_b < 1$. Hence, Lemma 2 always holds. ■

From Lemma 2, we obtain the following corollary.

Corollary 1: If $\omega_i > \omega_j$ and $p_j > 0$, then $p_i > p_j$.

Proof: If $\omega_i > \omega_j$ and $p_j > 0$, then (7) must hold for player i and player j, from which we conclude that $p_i > p_j > 0$. ■

In the following, we assume that the strategy profile of all players $\mathbf{p} = [p_1, p_2, \ldots, p_N]^\intercal$ is from a possible Nash equilibrium of the game. From Lemma 2, we know that $p_1 \geq \cdots \geq p_N \geq 0$ at any Nash equilibrium. We further assume that if $p_N = 0$, then m is the minimum index such that $p_m = 0$; otherwise, $m = N + 1$.

Consider the best response strategy profile of players that pay positive prices at a Nash equilibrium, denote it by $\{p_i \mid i < m\}$. According to (5), the strategy profile $\{p_i \mid i < m\}$ must satisfy

$$p_i + \frac{k_s}{k_b} \sum_{j \neq i} p_j + \frac{\mu}{k_b} = \sqrt{\frac{\omega_i}{\alpha k_b}}, \ \forall\ i < m. \tag{8}$$

Since $p_j = 0$ for $j \geq m$, from (8) we get

$$p_i + \frac{k_s}{k_b} \sum_{j < m, j \neq i} p_j + \frac{\mu}{k_b} = \sqrt{\frac{\omega_i}{\alpha k_b}}, \ \forall\ i < m. \tag{9}$$

The next lemma establishes a property of the solution of (9). With this property, we will be able to show that, given m, the strategy profile $\{p_i \mid i < m\}$ is unique.

Lemma 3: Equations (9) have a unique solution.

Proof: Equations (9) can be re-written as follows:

$$\begin{bmatrix} 1 & \frac{k_s}{k_b} & \cdots & \frac{k_s}{k_b} \\ \frac{k_s}{k_b} & 1 & \cdots & \frac{k_s}{k_b} \\ \vdots & & \ddots & \vdots \\ \frac{k_s}{k_b} & \frac{k_s}{k_b} & \cdots & 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{m-1} \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{\omega_1}{\alpha k_b}} - \frac{\mu}{k_b} \\ \sqrt{\frac{\omega_2}{\alpha k_b}} - \frac{\mu}{k_b} \\ \vdots \\ \sqrt{\frac{\omega_{m-1}}{\alpha k_b}} - \frac{\mu}{k_b} \end{bmatrix}. \tag{10}$$

We claim that (10) has a unique solution $\mathbf{q} = [p_1, p_2, \ldots, p_{m-1}]^\intercal$. Denote (10) by $\mathbf{Aq} = \mathbf{b}$. To prove our claim, we will show that the columns of $\mathbf{A}$ are linearly independent.

Denoting $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{m-1}]$, where $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{m-1}$ are columns vectors of $\mathbf{A}$, and denoting the ℓ-th item of vector $\mathbf{a}_i$ by $\mathbf{a}_i(\ell)$, we have

$$\begin{aligned} \mathbf{a}_i(\ell) &= 1, \ \ell = i, \\ \mathbf{a}_i(\ell) &= \frac{k_s}{k_b}, \ \ell \neq i. \end{aligned} \tag{11}$$

Assume by contradiction that the columns of $\mathbf{A}$ are linearly dependent. Then, there exists an i-th column that can be expressed as a linear combination of other columns:

$$\mathbf{a}_i = \sum_{j \neq i} \lambda_j \mathbf{a}_j, \ 1 \leq i, j \leq m - 1$$

which yields

$$\begin{aligned} \sum_{j \neq i} \lambda_j \mathbf{a}_j(i) &= \mathbf{a}_i(i), \\ \sum_{j \neq i} \lambda_j \mathbf{a}_j(\ell) &= \mathbf{a}_i(\ell), \ \forall\ \ell \neq i. \end{aligned} \tag{12}$$

Combining (11) with (12), we get

$$\sum_{j \neq i} \lambda_j \frac{k_s}{k_b} = 1, \tag{13}$$

$$\lambda_\ell + \sum_{j \neq i, j \neq \ell} \lambda_j \frac{k_s}{k_b} = \lambda_\ell \left(1 - \frac{k_s}{k_b}\right) + \sum_{j \neq i} \lambda_j \frac{k_s}{k_b} = \frac{k_s}{k_b}, \ \forall\ \ell \neq i. \tag{14}$$

By (13), and substituting the term $\sum_{j \neq i} \lambda_i k_s / k_b$ in (14) with 1, we get

$$(1 + \lambda_\ell) \left(1 - \frac{k_s}{k_b}\right) = 0, \ \forall\ \ell \neq i.$$

We already know that $0 < k_s/k_b < 1$, so it must be that

$$\lambda_\ell = -1, \ \forall\ \ell \neq i.$$

However, this contradicts (13), hence the assumption cannot hold. We thus conclude that the columns of $\mathbf{A}$ are linearly independent, which implies that $\mathbf{A}$ is invertible. Consequently, the solution $\mathbf{q} = \mathbf{A}^{-1}\mathbf{b}$ to (10) is unique, that is, equations (9) have a unique solution. ■

The next lemma helps to find the value m such that $p_i = 0$ for all $i \geq m$. Intuitively, a player is not willing to pay if there are enough free resources, which include shared resources and idle buy-in resources from players with larger workloads.

Lemma 4: At any Nash equilibrium of the SBC game, the best response strategy of player m is $p_m = 0$ if and only if

$$\frac{k_s}{k_b}\sum_{i<m} p_i^* + \frac{\mu}{k_b} \geq \sqrt{\frac{\omega_m}{\alpha k_b}}, \tag{15}$$

where $\{p_i^* \mid i < m\}$ is the unique solution to (9).

Proof: We will prove that (15) is a necessary and sufficient condition for $p_m = 0$ at a Nash equilibrium.

(A) *(Necessity):* $p_m = 0 \Rightarrow \frac{k_s}{k_b}\sum_{i<m} p_i^* + \frac{\mu}{k_b} \geq \sqrt{\frac{\omega_m}{\alpha k_b}}$.

From Lemma 2, at a Nash equilibrium, $p_m = 0$ implies $p_i = 0$ for $i \geq m$. Solving (9) yields $\{p_i^* \mid i < m\}$, which is the best response strategy profile for players $i < m$ in the case that $p_i = 0$ for $i \geq m$. Therefore, combining $\{p_i^* \mid i < m\}$ with $\{p_i = 0 \mid i \geq m\}$, we get a strategy profile for all players that is a Nash equilibrium. As a result, the best response strategy of player m is according to (5)

$$\begin{aligned} p_m &= \max\left(0, \sqrt{\frac{\omega_m}{\alpha k_b}} - \frac{k_s}{k_b}\sum_{i\neq m} p_i - \frac{\mu}{k_b}\right) \\ &= \max\left(0, \sqrt{\frac{\omega_m}{\alpha k_b}} - \frac{k_s}{k_b}\sum_{i<m} p_i^* - \frac{\mu}{k_b}\right) \\ &= 0, \end{aligned}$$

which implies

$$\frac{k_s}{k_b}\sum_{i<m} p_i^* + \frac{\mu}{k_b} \geq \sqrt{\frac{\omega_m}{\alpha k_b}}.$$

(B) *(Sufficiency):* $\frac{k_s}{k_b}\sum_{i<m} p_i^* + \frac{\mu}{k_b} \geq \sqrt{\frac{\omega_m}{\alpha k_b}} \Rightarrow p_m = 0$.

We will prove this by establishing its contraposition, namely:

$$p_m > 0 \Rightarrow \frac{k_s}{k_b}\sum_{i<m} p_i^* + \frac{\mu}{k_b} < \sqrt{\frac{\omega_m}{\alpha k_b}}.$$

Note that p_m is non-negative, hence the case $p_m < 0$ is not possible.

If $p_m > 0$, then the best response strategy of player m is according to (5)

$$p_m = \sqrt{\frac{\omega_m}{\alpha k_b}} - \frac{k_s}{k_b}\sum_{i\neq m} p_i - \frac{\mu}{k_b}. \tag{16}$$

Since $\{p_i^* \mid i < m\}$ is the solution to (9), we have

$$p_i^* + \frac{k_s}{k_b}\sum_{j<m, j\neq i} p_j^* = \sqrt{\frac{\omega_i}{\alpha k_b}} - \frac{\mu}{k_b}, \ \forall i < m. \tag{17}$$

According to the best response (5), the strategy profile $\{p_i \mid i < m\}$ at a Nash equilibrium must satisfy (8), which is equivalent to

$$p_i + \frac{k_s}{k_b}\sum_{j<m, j\neq i} p_j + \frac{k_s}{k_b}\sum_{j\geq m} p_j = \sqrt{\frac{\omega_i}{\alpha k_b}} - \frac{\mu}{k_b}, \ \forall i < m. \tag{18}$$

Summing (17) and (18) in terms of index i from 1 to $m-1$, respectively,

$$\begin{aligned} &\left(1 + \frac{k_s}{k_b}(m-2)\right)\sum_{i<m} p_i^* \\ &\quad = \sum_{i<m}\sqrt{\frac{\omega_i}{\alpha k_b}} - (m-1)\frac{\mu}{k_b}, \end{aligned} \tag{19}$$

$$\begin{aligned} &\left(1 + \frac{k_s}{k_b}(m-2)\right)\sum_{i<m} p_i + (m-1)\frac{k_s}{k_b}\sum_{j\geq m} p_j \\ &\quad = \sum_{i<m}\sqrt{\frac{\omega_i}{\alpha k_b}} - (m-1)\frac{\mu}{k_b}. \end{aligned} \tag{20}$$

We observe that the RHS of (19) is equal to the RHS of (20). Substituting the RHS of (19) with the LHS of (20) and re-arranging terms, we get

$$\begin{aligned} \sum_{i<m} p_i^* &= \sum_{i<m} p_i + \frac{(m-1)k_s/k_b}{(1+(m-2)k_s/k_b)}\sum_{j\geq m} p_j \\ &= \sum_{i<m} p_i + \frac{(m-1)k_s/k_b}{(1 - k_s/k_b + (m-1)k_s/k_b)}\sum_{j\geq m} p_j \\ &\leq \sum_{i<m} p_i + \sum_{j\geq m} p_j. \end{aligned}$$

Finally, from the above inequality and (16) we get

$$\begin{aligned} \frac{k_s}{k_b}\sum_{i<m} p_i^* + \frac{\mu}{k_b} &\leq \frac{k_s}{k_b}\left(\sum_{i<m} p_i + \sum_{j\geq m} p_j\right) + \frac{\mu}{k_b} \\ &= \frac{k_s}{k_b}\left(p_m + \sum_{i<m} p_i + \sum_{j>m} p_j\right) + \frac{\mu}{k_b} \\ &< p_m + \frac{k_s}{k_b}\sum_{i\neq m} p_i + \frac{\mu}{k_b} \\ &= \sqrt{\frac{\omega_m}{\alpha k_b}}, \end{aligned}$$

which completes the proof. ■

From Lemmas 3 and 4, we deduce that, once we find the minimum m such that $p_m = 0$, the solution set $\{p_i^* \mid i < m\}$ to (9) together with $\{p_i = 0 \mid i \geq m\}$ is the unique Nash equilibrium. This leads to Algorithm 1 for computing the Nash equilibrium of an N-player SBC game.

The following theorem formalizes this result.

Theorem 1: For an N-player SBC game, there exists a unique Nash equilibrium, which can be obtained by Algorithm 1.

Proof: If $\sqrt{\frac{\omega_1}{\alpha k_b}} \leq \frac{\mu}{k_b}$, then from Lemma 4, we know that $p_1 = 0$ is the best response strategy of player 1. Moreover, from Lemma 2, we know that $p_i \leq p_1 = 0$ for all $i > 1$, that is, for players $2, 3, \ldots, N$, the best response strategies are also 0. Thus, $\{0, 0, \ldots, 0\}$ is the unique Nash equilibrium of the N-player SBC game.

If $\sqrt{\frac{\omega_1}{\alpha k_b}} > \frac{\mu}{k_b}$, then we proceed to Steps $4 - 11$ of Algorithm 1 in order to find the Nash equilibrium. The

Algorithm 1: Computation of the Nash Equilibrium for an N-Player SBC Game

```
Output: $\{p_1, p_2, \ldots, p_N\}$
1  $i \leftarrow 1$;
2  $\{p_1, p_2, \ldots, p_N\} \leftarrow \{0, 0, \ldots, 0\}$;
3  if $\sqrt{\frac{\omega_1}{\alpha k_b}} > \frac{\mu}{k_b}$ then
4      while $i \leq N$ do
5          compute $\{p_1^*, p_2^*, \ldots, p_i^*\}$ by solving:
           $\begin{bmatrix} 1 & \frac{k_s}{k_b} & \cdots & \frac{k_s}{k_b} \\ \frac{k_s}{k_b} & 1 & \cdots & \frac{k_s}{k_b} \\ \vdots & & \ddots & \vdots \\ \frac{k_s}{k_b} & \frac{k_s}{k_b} & \cdots & 1 \end{bmatrix} \begin{bmatrix} p_1^* \\ p_2^* \\ \vdots \\ p_i^* \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{\omega_1}{\alpha k_b}} - \frac{\mu}{k_b} \\ \sqrt{\frac{\omega_2}{\alpha k_b}} - \frac{\mu}{k_b} \\ \vdots \\ \sqrt{\frac{\omega_i}{\alpha k_b}} - \frac{\mu}{k_b} \end{bmatrix}$;
6          $\{p_1, p_2, \ldots, p_i\} \leftarrow \{p_1^*, p_2^*, \ldots, p_i^*\}$;
7          if $i < N$ and $\frac{k_s}{k_b} \sum_{j \leq i} p_i^* + \frac{\mu}{k_b} \geq \sqrt{\frac{\omega_{i+1}}{\alpha k_b}}$ then
8              break;
9          end
10         $i \leftarrow i + 1$;
11     end
12 end
```

equations we need to solve in Step 5 at each iteration correspond to (9) in Lemma 3. As shown there, the solution to (9) is unique at each iteration.

Lemma 4 establishes that the best response strategy of player m is $p_m = 0$ if and only if (15) is satisfied. Thus, by going from $i = 1$ to $i = N$ and solving (9) iteratively until (15) is satisfied, we find all the $p_i > 0$, from which we get the index m such that $p_i > 0$ for $i < m$ and $p_i = 0, i \geq m$. In the end, we get a strategy profile from Algorithm 1, such that, for player $m, m+1, \ldots, N$, their best response strategies are 0; for player $1, 2, \ldots, m-1$, their best response strategies are per (9), hence the unique solution set $\{p_i^* \mid i < m\}$ to (9) constitutes the best response strategies for players $1, 2, \ldots, m-1$.

In conclusion, the output $\{p_1, p_2, \ldots, p_N\}$ of Algorithm 1, which is in the form of $\{p_1^*, p_2^*, \ldots, p_{m-1}^*, 0, \ldots, 0\}$, is a Nash equilibrium of the N-player SBC game. Moreover, since m is unique following Algorithm 1 and $\{p_1^*, p_2^*, \ldots, p_{m-1}^*\}$ is the unique solution set to (9), the Nash equilibrium $\{p_1, p_2, \ldots, p_N\}$ is unique. ■

In general, it is considered difficult (PPAD-complete) to find the Nash equilibrium for an N-player game [33]. However, for an SBC game, Theorem 1 shows that the Nash equilibrium is not only unique but also can be solved by Algorithm 1 in polynomial time, as established in the following.

Lemma 5: The Nash equilibrium of an N-player SBC game can be computed by Algorithm 1 in polynomial time, specifically in $O(N^4)$.

Proof: It takes $O(N)$ to initialize i and $\{p_1, p_2, \ldots, p_N\}$ (Steps $1-2$). Then, the algorithm first decides whether it needs to execute the while loop (Steps $4-11$). If it is executed, then, at each iteration, it takes $O(N^3)$ time to compute the best response equation (Step 5), $O(N)$ to update $\{p_1, p_2, \ldots, p_N\}$ (Step 6) as well as to check whether to break the loop (Steps $7-9$), and $O(1)$ to update i (Step 10). The "while" loop will be executed at most N times. In conclusion, the total running time of Algorithm 1 is upper-bounded by $O(N^4)$. ■

Next, we briefly analyze the efficiency of the unique Nash equilibrium of an SBC game. Define the social welfare as the sum of all players' payoffs

$$U(\mathbf{p}) = \sum_{i \in \Phi} U_i(\mathbf{p}) = \sum_{i \in \Phi} \left(\Gamma(\omega_i) - \frac{\omega_i}{\mu + k_b p_i + \sum_{j \neq i} k_s p_j} - \alpha p_i \right).$$

Consider each player i that pays a positive price $p_i > 0$ at the equilibrium. To maximize the player's own payoff $U_i(\mathbf{p})$, p_i has to satisfy

$$\frac{\partial U_i(\mathbf{p})}{\partial p_i} = \frac{k_b \omega_i}{\left(\mu + k_b p_i + \sum_{\ell \neq i} k_s p_\ell\right)^2} - \alpha = 0. \quad (21)$$

However, to maximize the social welfare, p_i has to satisfy

$$\frac{\partial U(\mathbf{p})}{\partial p_i} = \frac{k_b \omega_i}{\left(\mu + k_b p_i + \sum_{\ell \neq i} k_s p_\ell\right)^2} - \alpha + \sum_{j \neq i} \left(\frac{k_s \omega_j}{\left(\mu + k_s p_i + k_b p_j + \sum_{\ell \neq i, \ell \neq j} k_s p_\ell\right)^2} \right) = 0, \quad (22)$$

if the solution $p_i^\star$ to (22) is positive. Note that $\partial U_i(\mathbf{p})/\partial p_i < \partial U(\mathbf{p})/\partial p_i$, so if we substitute p_i in (21) with the solution $p_i^\star$ to (22), we get $\partial U_i(\mathbf{p})/\partial p_i^\star < 0$. Moreover, $\partial U_i(\mathbf{p})/\partial p_i$ is a strictly decreasing function of p_i, hence we get that the solution p_i to (21) is strictly smaller than the social optimal solution $p_i^\star$, and this holds for all players that pay positive prices at the Nash equilibrium. In other words, to maximize social welfare, each of these players should pay more. As a result, the Nash equilibrium of a shared/buy-in computing game is strictly sub-optimal, i.e., inefficient, in terms of the social welfare.

V. SBC GAMES AS AGGREGATIVE GAMES AND BEST-RESPONSE DYNAMICS

In this section, we establish a connection between SBC games and aggregative games. With that at hand, we are able to prove that an SBC game can always converge to its Nash equilibrium.

Consider a game $\langle \Phi, \{P_i\}_{i \in \Phi,}, \{U_i\}_{i \in \Phi} \rangle$ as defined in Section III, where Φ is the finite set of players, P_i is the non-empty strategy set of player i, and U_i is the payoff of player i. We say that a game $\langle \Phi, \{P_i\}_{i \in \Phi,}, \{U_i\}_{i \in \Phi} \rangle$ is an *aggregative game*, if for each player i, its payoff U_i is a function of p_i and $\sum_{j\in\Phi} p_j$, i.e., $U_i(\mathbf{p}) = \tilde{U}_i(p_i, \sum_{j\in\Phi} p_j)$. In an aggregative game, the payoff of player i depends only on its own strategy p_i and the aggregate of all players' strategies $\sum_{j\in\Phi} p_j$. As a result, it is enough for the player to know the aggregate in order to calculate its payoff, instead of the strategy of each specific player j.

Lemma 6: An SBC game is an aggregative game.

Proof: Note that in an SBC game, the payoff function U_i of player i can be written as

$$\begin{aligned} U_i(\mathbf{p}) &= \Gamma(\omega_i) - \frac{\omega_i}{\mu + k_b p_i + \sum_{j\neq i} k_s p_j} - \alpha p_i \\ &= \Gamma(\omega_i) - \frac{\omega_i}{\mu + (k_b - k_s)p_i + k_s \sum_{j\in\Phi} p_j} - \alpha p_i \\ &= \tilde{U}_i\left(p_i, \sum_{j\in\Phi} p_j\right). \end{aligned}$$

The lemma then follows. ■

Next, we show that SBC games have the property of strategic substitutes, or in other words, submodularity. That is, each player in the game will opt to pay a higher price if other players are paying lower prices, and vice versa.

Let p_{-i} denote the strategy of a player j other than i, we say that a payoff function $U_i(\mathbf{p})$ has decreasing difference in (p_i, p_{-i}) if for all $\tilde{p}_i \geq p_i$ and $\tilde{p}_{-i} \geq p_{-i}$,

$$U_i(\tilde{p}_i, \tilde{p}_{-i}) - U_i(p_i, \tilde{p}_{-i}) \leq U_i(\tilde{p}_i, p_{-i}) - U_i(p_i, p_{-i}).$$

Moreover, we say that the game $\langle N, \{P_i\}_{i\in\Phi,}, \{U_i\}_{i\in\Phi}\rangle$ is *submodular* if (a) P_i is a compact subset of $\mathbb{R}$; (b) the payoff function $U_i(\mathbf{p})$ is upper semi-continuous in (p_i, p_{-i}); (c) and the payoff function $U_i(\mathbf{p})$ has decreasing difference in (p_i, p_{-i}).

Lemma 7: An SBC game is submodular.

Proof: P_i is a compact subset of $\mathbb{R}$ since it is continuous by assumption, upper-bounded by $\sqrt{\frac{\omega_i}{\alpha k_b}} > 0$, and lower-bounded by 0, as shown in Section IV.

Given a twice continuously differentiable function $f : \mathbf{X} \to R$, f has decreasing difference in if and only if

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \leq 0, \ \forall\ i \neq j.$$

Consider the payoff function $U_i(p)$ of player i. We have

$$\frac{\partial^2 U_i(\mathbf{p})}{\partial p_i \partial p_j} = -\frac{k_b k_s \omega_i}{\left(\mu + k_b p_i + \sum_{\ell\neq i} k_s p_\ell\right)^3} < 0, \ \forall\, j \neq i.$$

Therefore, the payoff function $U_i(\mathbf{p})$ has decreasing difference in (p_i, p_{-i}). Furthermore, note that $\mu > 0$, $k_b > 0$, $k_s > 0$, $p_i \geq 0$ and $p_{-i} \geq 0$ for all i, from which we can get that $U_i(\mathbf{p})$ is continuous in (p_i, p_{-i}). Thus, we deduce that an SBC game is submodular. ■

Next, we will apply the results from [28] to our model, and show that an SBC game always converges to its Nash equilibrium through best response dynamics from all possible initial states.

Best response dynamics correspond to a procedure whereby, starting from an initial state, every player iteratively updates its strategy according to its best response (5). If an SBC game converges to a specific state through best response dynamics, then we know that every player must be playing its best response strategy, i.e., the Nash equilibrium is reached.

More specifically, we assume that best response dynamics acts in the following way: in each *round*, players update their strategies *once* one after another according to their best responses (5) given other players' strategies, and the process continues as long as an equilibrium is not reached.

Theorem 2: Through best response dynamics, an SBC game will converge to its unique Nash equilibrium from all possible initial states.

Proof: Firstly, we have proven that an SBC game is an aggregative game (Lemma 6). For each player i, its strategy set P_i is compact, and the payoff function $U_i(\mathbf{p})$ is continuous (as shown at the beginning of Section IV).

Secondly, the assumptions 1' and 2 of [28] are satisfied. Indeed, as explained in Remark 2 of [28], since the strategy sets P_i in our model are one-dimensional, Assumption 1' is essentially equivalent to submodularity which has already been proven (Lemma 7). Assumption 2 implies that the shift function for aggregating, which is a linear sum $\sum_{j\in\Phi} p_j$ in our model, exhibits strictly increasing differences in p_i and $\sum_{j\in\Phi, j\neq i} p_j$, possibly after a strict monotone transformation. As explained in Example 3 of [28], it can be shown that with a monotone transformation $h(\mathbf{z}) = \exp(\mathbf{z})$, the linear sum will exhibit strictly increasing differences.

Thirdly, best response dynamics will yield an admissible sequential improvement path defined in [28]. In each round, each player updates its strategy according to its best response, which implies that each player moves to a strictly preferred strategy if it exists, or stay with the previous strategy otherwise. Note that there is no indifference path in our model, since the best response (5) is single-valued. The path is admissible in the sense that each player gets the chance to move in each round, so that following the path everyone can keep updating until the Nash equilibrium is reached.

In conclusion, an SBC game satisfies all requirements of [28, Th. 2], thus will always converge to its unique pure Nash equilibrium through best response dynamics. Note that we have already proved the uniqueness of the pure Nash equilibrium. Moreover, the convergence was proven by considering limit conditions of best response dynamics, which implies the initial states have no influence to the convergence property. Hence, an SBC game always converges from all possible initial states. ■

Remark 4: In our model, we assume the aggregate to be a linear sum of all players' strategies, which simplifies the analysis and helps us establish the existence and uniqueness of the game's Nash equilibrium in a constructive manner. The results we get with the linear sum aggregate assumption have been validated using BU SCC data in Section VI. However, it might be worth allowing the aggregate to take a more general and complex form, so that it can apply to more scenarios. Note that, even if we relax the assumption about the aggregate, the convergence properties of the game still hold as long as the game is submodular and the aggregate exhibits a strictly increasing difference, possibly after a strict monotone transformation.

VI. NUMERICAL RESULTS

In this section, we provide numerical results to illustrate, confirm and expand on our theoretical results. We first describe and justify the setting of simulation parameters. Next, we thoroughly evaluate the dynamic behavior of the SBC game. We show that the game always converges, when players update their prices either sequentially (i.e., one by one) or in parallel. Next, we empirically evaluate the computational complexity of Algorithm 1 and indicate that it is indeed polynomial, and the order of growth is upper-bounded by 4, as stated by Lemma 5. Next, we verify Lemma 2, namely that prices paid by players are non-decreasing with their workloads, using actual data from the BU SCC. Last, we provide insight on the impact of model parameters on the resulting Nash equilibrium, and discuss design guidelines based on our analysis.

A. SIMULATION

In this subsection, we evaluate the results of Sections III and IV through simulation of best-response dynamics. We first discuss the simulation set-up.

1) SET-UP

We choose the BU SCC as a case study. The SCC has about 500 active projects running, so we simulate the best response dynamics of an SBC game with $N = 500$ players to investigate its behavior. In each *round*, each player updates its strategy *once* by playing its best response given the other players' strategies. We consider both the cases of sequential updates (i.e., players updates their strategies one after the other as in our definition of best response dynamics) and that of parallel updates, whereby all players update their strategies in parallel.

We classify users' jobs into three categories [7]: *shared* jobs are those running on shared nodes, *buy-in* jobs are those running on a user's own buy-in nodes, and *public* jobs are those running on other users' idle buy-in nodes. As of 2015-2016 [7], the shared workload on the BU SCC was 2.43×10^7 CPU-hours, the buy-in workload was 1.42×10^7 CPU-hours, and the public workload was 7.51×10^6 CPU-hours. We note that the aggregate shared, buy-in and public computing rates correspond respectively to $N\mu$, $k_b \sum p_i$, and $(N-1)k_s \sum p_i$, in our model.

Based on the workload characterization of [7], we assume that the project's workload (i.e., each players' job size in the game) is a random variable that follows a *log-normal* distribution. Specifically, let Ω denote the workload random variable, then

$$P(\Omega \leq \omega) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left[\frac{\ln \omega - \nu}{\sigma}\right], \tag{23}$$

where erf stands for the error function. In our simulation, we set $\nu = 7.37$ and $\sigma = 5.69$. In practice, any shared/buy-in computing systems has a maximum workload that it can sustain. In our simulation, we set that value to 5×10^6 CPU-hours. If a random sample exceeds that value, that sample is discarded. As a result, the median and mean of sample workloads are 906.87 and 1.77×10^5 CPU-hours, respectively.

We assume that the shared, buy-in and public nodes are used for approximately the same time in total, so that the ratio among workloads is approximately the same as the ratio among the computing rates reported above, i.e., $N\mu : k_b \sum p_i : (N-1)k_s \sum p_i \approx 24.3 : 14.2 : 7.5$. Therefore, we firstly set $\alpha = 1$ and $k_b = 30$, then $\mu = 775$ and $k_s = 0.0030$ accordingly to satisfy the ratio above. Note that the prices p_i that we obtain from simulation do not correspond exactly to how much players actually pay, but rather reflect their qualitative behavior (a different value of α would lead to different prices.)

2) NASH EQUILIBRIUM AND CONVERGENCE SPEED

Recall that best response dynamics is a procedure through which every player updates its strategy iteratively according to its best response (5). We investigate the best response dynamics of the SBC game following two updating rules: *sequential* updating and *parallel* updating. In the sequential updating case, players update their strategies one by one during each round. In the parallel updating case, players update their strategies at the same time during each round, which implies that each player makes its decision according to the strategy profile of all the other players in the previous round. Note that the sequential updating is exactly the same as the best response dynamics defined in Section V. The parallel updating case could represent the case where players update their strategies in a distributed manner. Since there may be some delay when inferring the latest strategies of other players, players simultaneously update their strategies in one round based on past information. In the following, each simulation is run 100 times, where each run uses a different random seed.

We first consider the case of an SBC game with 500 players and an "empty" initial state, i.e., the prices of all the players are initially set to 0. We assume that the game converges when the L2-norm of the players' strategy profile $\mathbf{p} = [p_1, p_2, \ldots, p_N]^\intercal$ changes by less than 10^{-6} in two consecutive rounds.

The simulations show that, in all the runs, the game converges to the Nash equilibrium predicted by Theorem 1. In the sequential updating case, the game converges to the Nash equilibrium within 6 ± 1 rounds (specifically within 6 rounds in 89% of the cases and 5 rounds in the remaining 11%); in the parallel updating case, the game converges within 11 ± 1 rounds (specifically within 10 rounds in 4% of the cases, 11 rounds in 91% of the cases, and 12 rounds in the remaining 5%), as also shown in Table 2. One of the

TABLE 2. Number of rounds needed for convergence from an empty initial state. The numbers in parenthesis correspond to the percentage of cases, based on 100 runs.

Updating rule	Sequential		Parallel		
Rounds	5 (11%)	6 (89%)	10 (4%)	11 (91%)	12 (5%)
Average	5.89		11.01		

TABLE 3. Number of rounds needed for convergence from an arbitrary initial state. The numbers in parenthesis correspond to the percentage of cases, based on 100 runs.

Updating rule	Sequential		Parallel	
Rounds	6 (10%)	7 (90%)	11 (26%)	12 (74%)
Average	6.90		11.74	

TABLE 4. Average number of rounds needed for convergence vs. number of players.

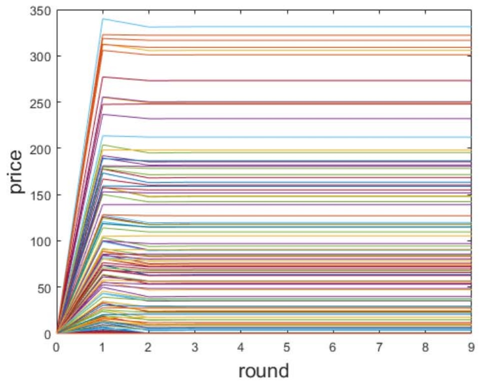
Number of players	200	300	400	500	600	700
Rounds (sequential)	6.53	6.67	6.80	6.90	6.93	6.99
Rounds (parallel)	11.33	11.55	11.62	11.74	11.78	11.81

simulation runs for the sequential updating case is illustrated in Fig. 1(a).

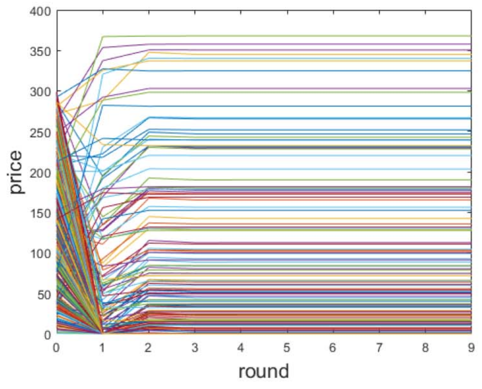
Next, we examine the convergence of the game's best response dynamics from an arbitrary initial state. We initialize the prices with random values that are uniformly distributed between 0 and 300. After 100 runs for both sequential and parallel updating, we find that the game always converges to the Nash equilibrium, although it generally takes slightly more rounds than from an empty initial state. In the sequential updating case, the game converges to the Nash equilibrium within 7 ± 1 rounds, while in the parallel updating case, the game converges with in 12 ± 1 rounds, as detailed in Table 3. One of the simulation runs for the sequential updating case is shown in Fig. 1(b), and one of the simulation runs for the parallel updating case is shown in Fig. 1(c). As expected, in all the cases above, the Nash equilibrium coincides with the output of Algorithm 1.

The fast convergence of the game may be explained as follows. The only term in the best response (5) of player i that changes between two consecutive rounds is $\frac{k_s}{k_b}\sum_{j\neq i} p_j$. If all players are not too far from the equilibrium, the difference $\Delta\frac{k_s}{k_b}\sum_{j\neq i} p_j$ will be relatively small compared with the best response p_i in the previous round. Moreover, since the best response of player i has two other unchanged terms $\sqrt{\frac{\omega_i}{\alpha k_b}}$ and $\frac{\mu}{k_b}$, one round of best response dynamics will bring every player not too far from the equilibrium (note that, in parallel updating case, players do not have the latest information so it takes two rounds). As a result, one can expect that after the second round, the players' strategies should not change much.

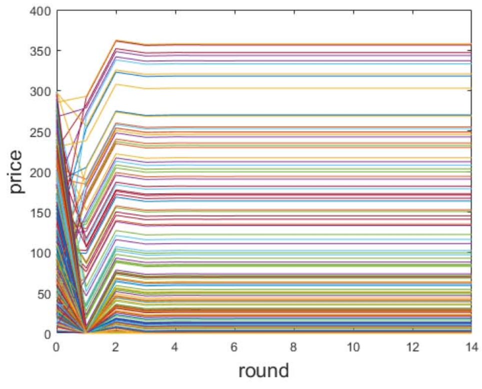
Next, we investigate the relationship between the convergence speeds (in terms of number of rounds till convergence) and the number of players. Again, we simulate the best response dynamics of the game from arbitrary initial states for 100 times. Detailed results for the both sequential and parallel updating cases are shown in Table 4. We find that the game converges fast, and the number of rounds required increases slowly with the number of players in the game.



(a) NE from empty initial state in sequential case.



(b) NE from arbitrary initial state in sequential case.



(c) NE from arbitrary initial state in parallel case.

FIGURE 1. Convergence of best response dynamics.

3) TIME COMPLEXITY

In this subsection, we investigate the time complexity for computing the Nash equilibrium. We compute the Nash equilibrium of the game with number of players ranging from $N = 400$ to $N = 6000$ in two ways: using best response dynamics from an empty initial state and with sequential updates, as done in the previous subsection, and computing the equilibrium using Algorithm 1. The details on the hardware and software used for this simulation are as follows: OS: Windows 10 Pro, CPU: AMD Ryzen 5 2600x 6-Core Processor, RAM: 16 GB, Software: MATLAB R2018b.

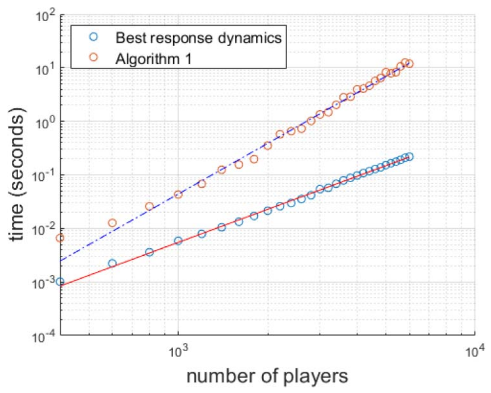


FIGURE 2. **Time complexity of computing the Nash equilibrium using (1) best response dynamics and (2) Algorithm 1.**

The corresponding results are shown in Fig. 2 on a logarithmic scale. We perform linear regression to the logarithms of the data, through which we empirically find that the time complexity of best response dynamics scales as ($N^{2.05}$), while the time complexity of Algorithm 1 scales as ($N^{3.14}$).

The time complexity of best response dynamics is as expected, namely: our simulations showed that the game converges in a nearly constant number of rounds. Considering that there are N players and it takes $O(N)$ to calculate one player's strategy according to best response (5) during each round, the overall time complexity should roughly be $O(N^2)$. However, it is hard to provide an exact result since we do not precisely know how many rounds are needed for convergence. We also note that the actual complexity of Algorithm 1 is below the upper bound of $O(N^4)$ provided by Lemma 5, hence the upper bound is slightly pessimistic. Regardless, using best response dynamics is a faster way to compute the Nash equilibrium.

4) INFLUENCE OF PARAMETERS

In this subsection, we choose three players out of the 500 players in the game as representatives, and change the values of the parameters $\mu, k_b, k_s, \{\omega_i\}$ in order to investigate the influence of these parameters on the game's Nash equilibrium. We set the job sizes of the chosen three players as $\{\omega_1, \omega_2, \omega_3\} = \{2 \times 10^5, 1 \times 10^5, 5 \times 10^4\}$. We scale each of the four parameters by a multiplicative factor β to evaluate the impact of the three players' strategies at the Nash equilibrium. Fig. 3 presents the results. We shall discuss these findings in the model analysis subsection.

B. VALIDATION OF RATIONAL BEHAVIOR USING BU SCC DATA

Similar to many prior works on game theory, our paper assumes that users are fully rational. Thus, the selected strategy of each user is based on the action that maximizes its payoffs, as provided by Eq. (5) in our case. This equation forms the basis of Lemma 2 and the rest of this paper, whereby the price paid by a user is non-decreasing with its workload.

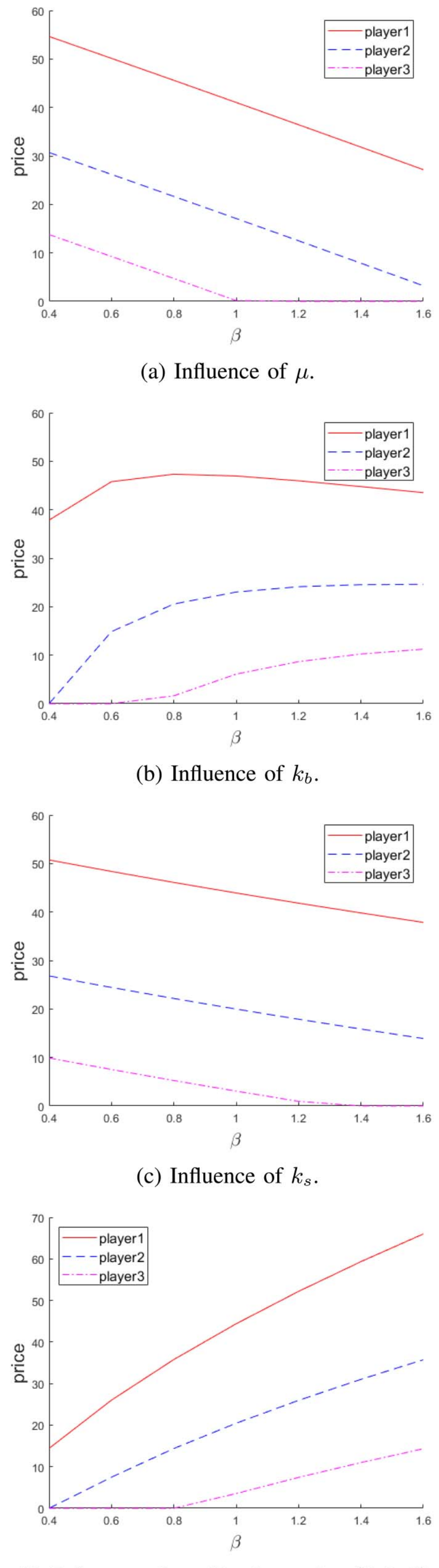


FIGURE 3. **Influence of parameters. β is a multiplicative scaling factor of the parameter under study.**

We next validate this assumption using BU SCC data collected from January 2019 to July 2019. As mentioned before, projects in BU SCC correspond to players in our

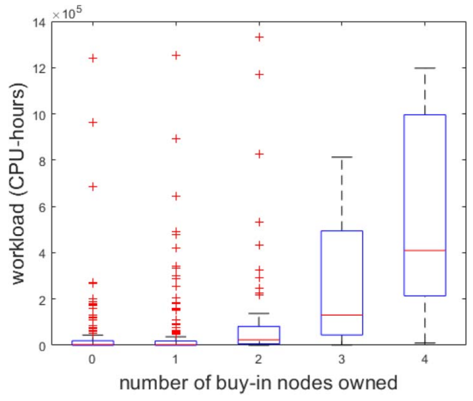


FIGURE 4. Number of buy-in nodes owned by a project vs. the workload of the project in the BU SCC [2].

TABLE 5. Number of buy-in nodes owned vs. average and median workload (in the units of cpu-hours) of projects in the BU SCC [2].

No. of buy-in nodes	0	1	2	3	4
Average workload	42035	76733	169250	383,818	565426
Median workload	3206.5	2638	23594	130556	409408

model. We investigate the relationship between the number of buy-in nodes owned by a project (which is commensurate with the price paid) and the project's total workload. A box plot is presented in Fig. 4, where the lowerbound and upperbound of a box represents the first and third quartile of the data, respectively, and the red line inside a box represents the median. The ends of the whiskers in the box plot represent the lowest and highest data still within 1.5 interquartile range (IQR) of the lower and upper quartile, respectively [34]. Table 5 shows detailed information about the average and median workloads. We generally observe a positive correlation between the number of buy-in nodes and the workload, which coincides with our basic assumption that a player with larger workload tends to pay more for buy-in resources.

We also note two interesting phenomena: (i) There are many "outliers" in Fig. 4, which implies that some projects are more price-sensitive than others (i.e., they have larger α). In other words, they rather endure longer job completion times than paying more; (ii) The players that own zero or one buy-in nodes have roughly the same amount of workload. Specifically, the average and median workloads of a project that owns zero buy-in node are 42035 CPU-hours and 3206.5 CPU-hours, respectively, while the average and median workloads of a project that owns one buy-in node are 76733 CPU-hours and 2638 CPU-hours, respectively. Indeed, it appears that some projects that own one buy-in node have little demand for it. However, this situation is rare when projects own at least two buy-in nodes.

C. MODEL ANALYSIS

1) INFLUENCE OF PARAMETERS

As established by Theorem 2, an N-player SBC game yields a unique Nash equilibrium, which can be explicitly calculated given the parameters $\mu, k_b, k_s, \omega_1, \omega_2, \ldots, \omega_N$. We next proceed to consider how these parameters influence the Nash equilibrium.

The parameters μ, k_b, k_s are set by the system provider. The parameter μ reflects the amount of shared resources. As expected, the larger μ is, the less players are willing to pay, as shown in Fig. 3(a). With a too large μ, no player will choose to buy in since they already have access to enough resources; with a too small μ, players are forced to buy in, but their costs may be too high, which may makes a shared/buy-in system not very attractive.

The parameter k_b measures how many resources a player can get by paying a price. A larger k_b implies that players get more buy-in resources for the same price. The influence of k_b is shown in Fig. 3(b), which is subtle and depends on the workload of each player. It is hard to tell whether a player will pay more or less as k_b increases, however, we do observe that players with lower workload will tend to pay more while players with larger workload will tend to pay less. We get similar insight by analyzing the players' best response strategies (see Eq. (5)): p_i is not a monotonic function of k_b, but for a player i with a larger workload ω_i, the derivative of its best response strategy p_i with respect to k_b is more likely to be negative, i.e., p_i is more likely to be a decreasing function of k_b, which explains why the price paid by player 1, who has the largest workload among the three players, starts decreasing earlier as the scaling factor β increases in Fig. 3(b).

The parameter k_s measures how much a player's buy-in resources can benefit other players. A larger k_s implies that a player can make more use of other players' buy-in resources. The influence of k_s is shown in Fig. 3(c). As k_s increases, users can access more "public" resources made available by idle buy-in nodes, and hence are less likely to purchase their own buy-in resources. This can also be explained by analyzing players' best response strategies (see Eq. (5)): note that $p_i \geq 0$ for all i, thus p_i is always a non-increasing function of k_s. Another interesting result is that, as k_s increases, p_i decreases at a slower speed since the absolute value of the derivative of p_i with respect to k_s, which is $\sum_{j \neq i} p_j / k_b$, gets smaller.

The set $\{\omega_1, \omega_2, \ldots, \omega_N\}$ corresponds to the workloads of the players. Given μ, k_b, k_s, as ω_i gets larger, player i needs more resources and will tend to pay more for buy-in resources, as indicated by Lemma 2. How a player perceives the parameters μ, k_b, k_s is also relevant to its workload. When μ, k_b, k_s changes, all player strategies will change accordingly, however, the strategy of a player with a larger workload will change relatively less. The influence of ω_i is shown in Fig. 3(d).

2) POTENTIAL TRAPS AND DESIGN GUIDELINES

Define players that pay strictly positive prices, i.e., $p_i > 0$, as "heavy users", and players that do not pay, i.e., $p_i = 0$, as "light users". Heavy users have to buy in resources in order to minimize their costs. Although heavy users benefit each other through buy-in resources, users with larger workloads benefit relatively less than users with smaller workloads. For example, consider two heavy users with workloads ω_1 and ω_2, where $\omega_1 > \omega_2$, then user 1 gets additional computing rate $k_s p_2$ from user 2, while user 2 gets additional computing rate $k_s p_1$ from user 1. From Lemma 2, we know that $p_1 > p_2$, as a result, user 1 gets relatively less additional resources even though it needs more resources due to its larger workload. On the other hand, shared resources and buy-in resources from other users can satisfy the needs of light users. Hence their best response strategy is to pay nothing, which implies that light users benefit the most from the shared/buy-in feature.

Thus, light users behave as "free-riders" in the system. In essence, this game incorporates some inherent unfairness, since a user with smaller workload relatively experiences a larger benefit. Moreover, a heavy user may be tempted to lower its cost by splitting its job into $M > 1$ shares, and then join the system as M light users, each being a free-rider. A possible solution for preventing users from such an artificial split of jobs is to charge all users of the system (e.g., in the University of Illinois at Urbana-Champaign Campus Cluster [35], light users still need to pay for shared resources). Another solution is to check the users' identity before being admitted into the system. For example, the SCC at Boston University can only be accessed by valid faculty and research staff members, hence a user cannot (easily) split its job in this system.

VII. CONCLUSION

We proposed a game-theoretic model for shared/buy-in computing systems, which provides a formal method to analyze the behavior of such systems. We investigated both static and dynamic properties of the game. We established that the game admits a unique Nash equilibrium, and, furthermore, we provided a polynomial-time algorithm for computing it. We then established that, regardless of the initial state, the game converges to the Nash equilibrium through best response dynamics. This result was obtained by showing that a SBC game belongs to the class of aggregative games.

Through numerical simulations, we explored additional convergence properties of the game and the effect of the system parameters on the structure of the Nash equilibrium. In particular, we found that the Nash equilibrium can be computed faster using best response dynamics than through Algorithm 1. However, only Algorithm 1 provably provides the exact answer in a deterministic amount of time. Another interesting insight from the paper is that increasing the amount of buy-in resources that a user gets per currency unit (i.e., increasing the parameter k_b) may lead some users to pay less and other users to pay more. We also investigated potential traps in the design of SBC systems, including the possible emergence of free-riders, and suggested corresponding guidelines for addressing this issue.

Our work is an initial attempt to formally understand the behavior of practical shared/buy-in computing systems. As such, it opens several interesting directions for future research, in particular: (i) investigating the perspective of social welfare, most notably the price of anarchy of the game; (ii) adding the service provider to the game, including its objective and strategic behavior; (iii) relaxing the assumption of a linear sum within the cost function into a more general aggregate form, in order to generalize our results. One of the goals of shared/buy-in systems from a provider's perspective is to consolidate IT services (all the users in an organization use the same platform). A system where light users would be charged for resource usage (which may appear fairer/more efficient) may deter such users from entering the system in the first place and defeat to some extent the purpose of the system.

REFERENCES

[1] (2019). *Shared Computing Cluster*. [Online]. Available: https://hadoop.apache.org/

[2] (2019). *Shared Computing Cluster*. [Online]. Available: https://www.bu.edu/tech/support/research/computing-resources/scc/

[3] (2019). *Research Computing*. [Online]. Available: https://its.northeastern.edu/services/research-computing/

[4] (2019). *HPC Cluster Basic Use Guide*. [Online]. Available: http://chtc.cs.wisc.edu/HPCuseguide.shtml

[5] (2019). *High Performance Computing*. [Online]. Available: https://it.arizona.edu/service/high-performance-computing

[6] (2019). *Condo Cluster Service*. [Online]. Available: http://research-it.berkeley.edu/services/high-performance-computing/condo-cluster-service

[7] C. Liao, Y. Klausner, D. Starobinski, E. Simhon, and A. Bestavros, "A case study of a shared/buy-in computing ecosystem," *Clust. Comput.*, vol. 21, no. 3, pp. 1595–1606, 2018.

[8] L. C. Corchón, "Comparative statics for aggregative games the strong concavity case," *Math. Soc. Sci.*, vol. 28, no. 3, pp. 151–165, 1994.

[9] P. Dubey, O. Haimanko, and A. Zapechelnyuk, "Strategic complements and substitutes, and potential games," *Games Econ. Behav.*, vol. 54, no. 1, pp. 77–94, 2006.

[10] M. Calzarossa, L. Massari, and D. Tessera, "Workload characterization issues and methodologies," in *Performance Evaluation: Origins and Directions*. Heidelberg, Germany: Springer, 2000, pp. 459–482.

[11] S. Di, D. Kondo, and W. Cirne, "Characterization and comparison of cloud versus grid workloads," in *Proc. IEEE Int. Conf. Clust. Comput.*. 2012, pp. 230–238.

[12] E. A. Meirom, S. Mannor, and A. Orda, "Strategic formation of heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 751–763, Mar. 2017.

[13] D. Niyato and E. Hossain, "Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2008–2017, May 2009.

[14] E. Kavurmacioglu, M. Alanyali, and D. Starobinski, "Competition in private commons: Price war or market sharing?" *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 29–42, Feb. 2016.

[15] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Comput. Surveys*, vol. 45, no. 3, p. 25, 2013.

[16] I. Abraham, D. Dolev, R. Gonen, and J. Halpern, "Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation," in *Proc. 25th Annu. ACM Symp. Princ. Distrib. Comput.*, 2006, pp. 53–62.

[17] J. Londoño, A. Bestavros, and S.-H. Teng, "Colocation games: And their application to distributed resource management," in *Proc. USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, San Diego, CA, USA, Jun. 2009, p. 10.

[18] E. Simhon and D. Starobinski, "Advance reservation games," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 2, no. 2, pp. 1–21, Apr. 2017.

[19] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 954–1001, 2nd Quat., 2017.

[20] V. Abhishek, I. A. Kash, and P. Key, "Fixed and market pricing for cloud services," in *Proc. IEEE INFOCOM Workshops*, 2012, pp. 157–162.

[21] A. Nahir, A. Orda, and D. Raz, "Workload factoring: A game-theoretic perspective," *IEEE/ACM Trans. Netw.*, vol. 23, no. 6, pp. 1998–2009, Dec. 2015.

[22] D. M. Topkis, "Equilibrium points in nonzero-sum *n*-person submodular games," *SIAM J. Control Optim.*, vol. 17, no. 6, pp. 773–787, 1979.

[23] J. Levin, "Supermodular games," Lectures Notes, Dept. Econ., Stanford Univ., Stanford, CA, USA, 2003.

[24] J. Huang, R. A. Berry, and M. L. Honig, "Distributed interference compensation for wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1074–1084, May 2006.

[25] D. D. Yao, "*S*-modular games, with queueing applications," *Queueing Syst.*, vol. 21, nos. 3–4, pp. 449–475, 1995.

[26] E. Altman and Z. Altman, "S-modular games and power control in wireless networks," *IEEE Trans. Autom. Control*, vol. 48, no. 5, pp. 839–842, May 2003.

[27] M. Dindoš and C. Mezzetti, "Better-reply dynamics and global convergence to Nash equilibrium in aggregative games," *Games Econ. Behav.*, vol. 54, no. 2, pp. 261–292, 2006.

[28] M. K. Jensen, "Aggregative games and best-reply potentials," *Econ. Theory*, vol. 43, no. 1, pp. 45–66, 2010.

[29] M. Voorneveld, "Best-response potential games," *Econ. Lett.*, vol. 66, no. 3, pp. 289–295, 2000.

[30] R. Cornes and R. Hartley, "Aggregative public good games," *J. Public Econ. Theory*, vol. 9, no. 2, pp. 201–219, 2007.

[31] R. Cornes, "Aggregative environmental games," *Environ. Resource Econ.*, vol. 63, no. 2, pp. 339–365, 2016.

[32] J. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.

[33] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," *SIAM J. Comput.*, vol. 39, no. 1, pp. 195–259, 2009.

[34] G. Upton and I. Cook, *Understanding Statistics*. Oxford, U.K.: Oxford Univ. Press, 1996.

[35] (2019). *Buy Compute*. [Online]. Available: https://campuscluster.illinois.edu/access/buy-compute/