The Proportional-share Allocation Market for Computational Resources*

Michal Feldman† Kevin Lai‡ Li Zhang‡

Submission to IEEE Transactions on Parallel and Distributed Systems

Abstract

We study the problem of allocating shared resources, such as bandwidth in computer networks and computational resources in shared clusters, among multiple users by the *proportional-share market mechanism*. Under this mechanism, each user partitions his budget among the multiple resources and receives a fraction of each resource proportional to his bid. We first formulate the resource allocation game under the proportional-share mechanism and study the efficiency and fairness of the equilibrium in this game. We present analytic and simulation results demonstrating that the proportional-share mechanism achieves a reasonable balance of high degrees of efficiency and fairness at the equilibrium.

Key words: game theory, resource allocation, proportional-share mechanism

1 Introduction

Sharing of resources can significantly increase a system's throughput by using statistical multiplexing to exploit the bursty utilization pattern of typical users. Some examples of resources that benefit from this type of sharing are network capacity and computational resources of shared clusters (e.g., the Grid [9] and PlanetLab [3]).

*An extended abstract was presented at the ACM Conference on Electronic Commerce, 2005

†School of Business Administration and Center for the Study of Rationality, Hebrew University of Jerusalem. Jerusalem 91904, Israel. michal.feldman@huji.ac.il. This work was partially conducted while at the School of Information at the University of California at Berkeley and at HP Labs.

‡HP Labs, 1501 Page Mill Road, Palo Alto, CA 94304. {klai,l.zhang}@hp.com

One of the challenges in such systems is how to allocate the shared resources among the different users both efficiently and fairly. Traditionally, the sharing is done through polices such as first come first serve and round-robin time sharing. Such approach has very small overhead but it lacks the capability for the user to express the priority of their tasks. It becomes more problematic as the different components of such systems are often operated by autonomous individuals and companies with diverse economic interests.

One promising alternative for allocating resource in computer systems is to employ a market-based approach. In a market-based approach, users submit bids for resources, and the resources are allocated according to some pre-defined resource allocation mechanisms. Previous work analyzed the efficiency and fairness of various market-based allocation mechanisms (some examples are [2, 5, 8, 22, 30, 32]). In this paper, we study a particular scheme in which each resource, such as CPU time, is treated as a divisible good and is allocated by the *proportional-share* allocation mechanism.

In the proportional-share allocator, each user submits bids for the different resources and receives a fraction of each resource equal to his bid divided by the sum of all the bids submitted for that resource. Compared to the other market mechanisms, such as auction based methods, this scheme is simple, computationally efficient, and scalable. In general, few market mechanisms for computation have practical implementions and fewer are in operation. Nonetheless, the proportional share allocator has been implemented [5, 16] and Tycoon [16, 1] is still in use (as of 2008). In particular, Tycoon is in operation at HP Labs, the European Organization for Nuclear Research (CERN), and various sites of the grid organization Enabling Grids for E-scienceE (EGEE). This degree of deployment is due the ease of implementing proportional-share on top of the existing proportional-share allocators that are commonly available in operating systems and virtual machine systems [31, 28, 19, 21, 6].

In this paper, we investigate the efficiency and fairness of the proportional-share allocation mechanism, with a focus on the linear utility function model, though we also consider non-linear models. Although the linear model sacrifices some fidelity, it gains considerable tractability. In addition, most users find it difficult to specify non-linear models.

In our model, m users, each with a budget, compete on n parallel resources, e.g. network bandwidth, CPU cycles, memory, or storage of hosts in networked shared clusters. The private preferences of each user are given by a vector of weights for each resource, reflecting the individual valuations for the different resources. Each user partitions his budget among the different resources and receives a fraction of each resource equal to his bid divided by the sum of all the bids submitted for that resource. In our analysis, we consider the case where each user has a linear utility function in terms of the amount of resource received, i.e. each user's utility is a weighted sum of the fraction the user gets from each resource. This assumption is consistent with the practice in which one is usually most concerned with the total number of CPU cycles or the total size of the storage space received.

We model the problem as a game in which each user tries to maximize his own utility function by dividing the budget wisely. The central questions we would like to investigate are the existence and the quality of the Nash equilibria of this game. Using game theoretic analysis to analyze the performance of systems has become a very common approach in recent years. This is in part due to the emergence of the Internet and many other distributed systems, which are composed of distributed computer networks managed by multiple administrative authorities and shared by users with competing interests. As such, they can be modeled as non-cooperative games. Our game theoretic analysis takes this approach. Thus, rather than developing techniques to compute the optimal outcome, we evaluate the quality of the stable outcomes in comparison to the optimal one.

For evaluating the Nash equilibria, we consider both their efficiency and fairness. Efficiency is the ratio of the achieved social welfare and the social optimum, also known as the *price of anarchy* [14]. Fairness in our game is measured by two metrics; namely, *utility uniformity* and *envy-freeness*. Utility uniformity measures the closeness of utilities of different users, while envy-freeness measures the happiness of users with their own resources compared to the resources of others.

1.1 Our results:

- **Analysis of the existence and quality of Nash equilibria.** We show analytically that there always exists a Nash equilibrium in the bounded budget game if the utility functions satisfy a fairly weak and natural condition of *strong competitiveness*. We also show the worst case performance bounds: for m players the efficiency at equilibrium is $\Omega(1/\sqrt{m})$ and the utility uniformity is $\geq 1/m$.

- **Design of algorithms for utility maximization.** We show that strategic users with linear utility functions can calculate their bids using a *best response* algorithm that quickly results in an allocation with high efficiency with little computational and communication overhead. We present variations of the best response algorithm for both finite and infinite parallelism tasks. In addition, we present a *local greedy adjustment* algorithm that converges more slowly than best response, but allows for non-linear or unformulatable utility functions.

- **Simulation results of efficiency and fairness.** We find that although the socially optimal allocation results in perfect efficiency, it may result in very poor fairness. Likewise, allocating according to only users' preference weights results in a high fairness, but a mediocre efficiency. Intuition would suggest that efficiency and fairness are exclusive. Surprisingly, simulation results indicate that the Nash equilibrium, reached by each user iteratively applying the best response algorithm to adapt his bids, achieves nearly the efficiency of the social optimum and nearly the fairness of the weight-proportional allocation: the efficiency is ≥ 0.90, the utility uniformity is ≥ 0.65, and the envy-freeness is ≥ 0.97, independent of the number of users in the system. In addition, the time to converge to the equilibrium is ≤ 5 iterations when all users use the best response strategy. The local adjustment algorithm performs similarly when there is sufficient competitiveness, but takes 25 to 90 iterations to stabilize.

Related work. The proportional share mechanism has been studied in the context of relating non-price-taking and price-taking behaviors in exchange economies. Shapley and Shubik proposed the *trading post game* model [27] to "explore the transition zone between perfect competition and oligopolistic competition". In the game, each trader places a monetary bid on each good held at a trading post and receives each good at a fraction of his bid to the total bid placed on that good, hence the "proportional response" from each trading post. This mechanism has since become the building block of non-price-taking models for economies with divisible goods.

In particular, it has been used as price-anticipating allocation schemes in the context of allocation of network capacity for flows [10, 11, 12, 13, 26]. One important assumption made in those existing works is that there is no budget constraint on each player. This assumption allows one to relate the Nash equilibrium to the solution of a (usually convex) global optimization problem, similar to the work of congestion (potential) games [18, 25]. However, those techniques no longer apply to our game because we model users as having bounded budgets and private preferences for resources. For example, unlike those games, our game may admit multiple Nash equilibria. Milchtaich [17] studied congestion games with private preferences but the technique in [17] is specific to a congestion game. Thus, by introducing the natural constraint of bounded budget, our model results in a qualitatively different game and requires different techniques.

Organization. The rest of the paper is organized as follows. We describe the model in Section 2. In section 3 we derive the existence of Nash equilibrium in our game, and sections 4 and 5 analyze the efficiency and fairness of the equilibrium allocation. In Section 6, we describe algorithms for users to optimize their own utility in the bounded budget game. In Section 7, we describe our simulator and simulation results. We conclude by discussing some limits of our model and future work in Section 8.

2 Model and Preliminaries

We study the problem of allocating a set of divisible resources. We refer to the resources as machines, although our model and analysis apply to the allocation of any type of computational resource. Suppose that there are m users and n machines. Each machine can be continuously divided for allocation to multiple users. An *allocation scheme* $\omega = (\mathbf{r}_1, \ldots, \mathbf{r}_m)$, where $\mathbf{r}_i = (r_{i1}, \cdots, r_{in})$ with r_{ij} representing the share of machine j allocated to user i, satisfies that for any $1 \leq i \leq m$ and $1 \leq j \leq n$, $r_{ij} \geq 0$ and $\sum_{i=1}^{m} r_{ij} \leq 1$. Let Ω denote the set of all the allocation schemes.

Resource Allocation Scheme. We consider the *price anticipating mechanism* in which each user submits a bid to each machine, and the price of the machine is determined by the total bids submitted. Formally, suppose that user i submits a non-negative bid x_{ij} to machine j. The price of machine j is then set to $Y_j = \sum_{i=1}^{n} x_{ij}$, the total bids placed on the machine j. The allocation of the resources is done according to the proportional share mechanism. That is, user i receives a fraction of $r_{ij} = \frac{x_{ij}}{Y_j}$ of j. When $Y_j = 0$, i.e. when there is no bid on a machine, the machine is not allocated to anyone. We call $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})$ the bidding vector of user i.

The additional consideration we have is that each user i has a budget (or money) constraint X_i. Therefore, user i's total bids have to sum up to his budget, i.e. $\sum_{j=1}^{n} x_{ij} \leq X_i$. The budget constraints come from the fact that the users do not have infinite budget, which is a standard assumption in economic models.

Utility Functions. Each user i's utility is represented by a function U_i of the fraction $(r_{i1}, \ldots, r_{in})$ the user receives from each machine, and the vector $(w_{i1}, \ldots, w_{in})$ of weights he assigns to each machine, reflecting his private preferences for the different machines. Given the problem domain we consider, the basic utility function we consider is the *linear utility function*: $U_i(r_{i1}, \cdots, r_{in}) = w_{i1}r_{i1} + \cdots + w_{in}r_{in}$, where $w_{ij} \geq 0$ is user i's private preference, also called his *weight*, on machine j. For example, suppose machine 1 has a faster CPU but less memory than machine 2, and user 1 runs CPU bounded applications, while user 2 runs memory bounded applications. As a result, $w_{11} > w_{12}$ and $w_{21} < w_{22}$.

We remark that in our model, the money serves as an intermediary and bears no intrinsic utility to the user, and the user should burn all the budget at the optimum bidding. While this is the standard treatment in economics, for examples in the pari-mutuel [7] and Fisher market model [4], one may assume that money has intrinsic value because of its potential of bringing utility to the user in future. In this case, we may consider the quasilinear utility function with the form $u_i(r_{i1}, \cdots, r_{in}; b_i) = \sum_j w_{ij}r_{ij} + b_i$, where b_i is the amount of budget left to i after he places his bids. Such a quasilinear utility function can be approximated by a linear utility function as follows. Take a number $B \gg \sum_i X_i$ and $\gg \sum_{i,j} w_{ij}$. We add an artificial resource m_0 and a buyer 0 with the properties that the buyer 0 has a budget of B and is solely interested in m_0. We then let $u'_i(r_{i0}, r_{i1}, \cdots, r_{in}) = Br_{i0} + \sum_j w_{ij}r_{ij}$. It can be easily verified that the economy with utility functions u' approximates that with quasilinear utility functions u within any degree when we choose a sufficiently large B. Or intuitively, m_0 is a resource from which the users may get approximately constant return of their bids. We hence only focus on the model of linear utility functions in this paper.

Another implication of the linear utility functions is that each user has enough jobs or enough parallelism within jobs to utilize all the resources he obtains. Consequently, the user's goal is to grab as much of a resource as possible. We call this the *infinite parallelism model*. In practice, a user's application may have an inherent limit on parallelization (e.g., some computations must be done sequentially) or there may be a system limit (e.g., the application's data is being served from a file server with limited capacity). To model this, we also consider the more realistic *finite* *parallelism* model, where the user's parallelism is bounded by k_i, and the user's utility U_i is the sum of the k_i largest $w_{ij}r_{ij}$. In this model, the user only submits bids to up to k_i machines. Our abstraction is to capture the essence of the problem and facilitate our analysis. In Section 8, we discuss the limit of the above definition of utility functions.

Best Response. As typically, we assume the users are selfish and strategic — they all act to maximize their own utility, defined by their utility functions. From the perspective of user i, if the total bids of the other users placed on each machine j is y_j, then the *best response* of user i to the system is the solution of the following optimization problem:

$$\text{maximize } U_i\left(\frac{x_{i1}}{x_{i1}+y_1}, \ldots, \frac{x_{in}}{x_{in}+y_n}\right) \text{ subject to}$$

$$\sum_{j=1}^{n} x_{ij} = X_i, \text{ and } x_{ij} \geq 0.$$

The difficulty of the above optimization problem depends on the formulation of U_i. In Section 6, we treat the algorithmic aspects of computing the best response in a dynamic setting. We show how to solve it for the infinite parallelism model and provide a heuristic for finite parallelism model.

Nash Equilibrium. A *Nash equilibrium* is a state (i.e., collection of bidding vectors), one for each user, such that each user's bidding vector is the best response to those of the other users. Formally, the bidding vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ form a *Nash equilibrium* if for any $1 \leq i \leq m$, $\mathbf{x}_i$ is the best response to the system, or, for any other bidding vector $\mathbf{x}'_i$,

$$U_i(\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_m) \geq U_i(\mathbf{x}_1, \ldots, \mathbf{x}'_i, \ldots, \mathbf{x}_m)\,.$$

The Nash equilibrium is desirable because it is a stable state at which no one has incentive to change his strategy. Yet, not all games admit a Nash equilibrium. Indeed, a Nash equilibrium may not exist in the price anticipating scheme we define above. This can be shown by a simple example of two players and two machines. For example, let $U_1(r_1, r_2) = r_1$ and $U_2(r_1, r_2) = r_1 + r_2$. Then player 1 should never bid on machine 2 because it has no value to him. Now, player 2 has to put a positive bid on machine 2 to claim the machine, but there is no lower limit to his bid, resulting in the non-existence of the Nash equilibrium. We should note that even a mixed strategy equilibrium does not exist in this example (which does not contradict Nash's existence theorem since the action space is infinite in our case). Clearly, this happens whenever there is a resource that is "wanted" by only one player. To rule out this case, we consider those *strongly competitive* games.[1] Under the infinite parallelism model, a game is called *strongly competitive* if for any $1 \leq j \leq n$, there exists an $i \neq k$ such that $w_{ij}, w_{kj} > 0$.

In cases where a Nash Equilibrium exists, the next important question is the performance at the Nash equilibrium, which is often measured by its efficiency and fairness. For a meaningful discussion of efficiency and fairness, we assume that the users are symmetric by requiring that $X_i = 1$ and $\sum_{j=1}^{n} w_{ij} = 1$ for all the $1 \leq i \leq m$. Informally, we require that all the users will have the same budget and the same utility when they own all the resources. This precludes the case in which a user with an extremely high budget has low valuations for the resources, resulting in very low efficiency or low fairness at equilibrium.

Efficiency. For an allocation scheme $\omega \in \Omega$, denote by $U(\omega) = \sum_i U_i(\mathbf{r}_i)$ the social welfare under ω. Let $U^* = \max_{\omega \in \Omega} U(\omega)$ denote the optimal social welfare — the maximum possible aggregated user utilities. The efficiency at an allocation scheme ω is defined as $\pi(\omega) = \frac{U(\omega)}{U^*}$. Let Ω_0 denote the set of the allocation at the Nash equilibrium. When there exists Nash equilibrium, i.e. $\Omega_0 \neq \emptyset$, define the *efficiency* of a game Q to be $\pi(Q) = \min_{\omega \in \Omega_0} \pi(\omega)$.

It is usually the case that $\pi < 1$, i.e. there is an efficiency loss at a Nash equilibrium. Papadimitriou [20] coined the term "price of anarchy", defined as the ratio between the social optimum and the worst case Nash equilibrium *thus it is always ≥ 1). Informally, the price of anarchy is the price incurred due to the fact that each user acts independently with his own objective function in mind. In cases where the price of anarchy is relatively small (i.e., close to 1), it serves as an indication for robustness against selfish behavior. Our efficiency measure is the inverse of the price of anarchy, and is always ≤ 1. In our case, a high value (i.e. close to 1) indicates a high efficiency level.

Fairness. While the definition of efficiency is standard, there are multiple ways to define fairness. We consider two metrics. One is by comparing the users' utilities. The *utility uniformity* $\tau(\omega)$ of an allocation scheme ω is defined to be $\frac{\min_i U_i(\omega)}{\max_i U_i(\omega)}$, the ratio of the minimum utility and the maximum utility among the users. Such definition (or utility discrepancy defined similarly as $\frac{\max_i U_i(\omega)}{\min_i U_i(\omega)}$) is

[1] Alternatives include adding a reservation price or limiting the lowest allowable bid to each machine. These alternatives, however, introduce the problem of coming up with the "right" price or limit.

used extensively in Computer Science literature. Under this definition, the utility uniformity $\tau(Q)$ of a game Q is defined to be $\tau(Q) = \min_{\omega \in \Omega_0} \tau(\omega)$.

The other metric, extensively studied in Economics, is the concept of envy-freeness [29]. Unlike the utility uniformity metric, the envy-freeness concerns how the user perceives the value of the share assigned to him, compared to the shares other users receive. Within such a framework, define the *envy-freeness* of an allocation scheme ω by $\rho(\omega) = \min_{i,j} \frac{U_i(\mathbf{r}_i)}{U_i(\mathbf{r}_j)}$. When $\rho(\omega) \geq 1$, the scheme is known as an envy-free allocation scheme. Likewise, the envy-freeness $\rho(Q)$ of a game Q is defined to be $\rho(Q) = \min_{\omega \in \Omega_0} \rho(\omega)$.

Best-Response Dynamics. Best-response dynamics describe dynamics in which each player, in its turn, responds with the action that yields him he maximum utility, given the strategies taken by the other players. In some games, best-response dynamics are guaranteed to converge to a NE, while in other games, best-response dynamics do not necessarily converge. In this paper, we explore the algorithmic aspects of best-response computation, as well as the question of convergence.

3 Nash Equilibrium Existence

We first provide a characterization of the equilibria. By definition, the bidding vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ is a Nash equilibrium if and only if each player's strategy is the best response to the group's bids. Since U_i is a linear function and the domain of each users bids $\{(x_{i1}, \ldots, x_{in}) | \sum_j x_{ij} = X_i \text{ , and } x_{ij} \geq 0\}$ is a convex set, the first order optimality condition, by applying Lagrange multiplier, is that there exists $\lambda_i > 0$ such that

$$\frac{\partial U_i}{\partial x_{ij}} = w_{ij} \frac{Y_j - x_{ij}}{Y_j^2} \begin{cases} = & \lambda_i \quad \text{if } x_{ij} > 0\text{, and} \\ \leq & \lambda_i \quad \text{if } x_{ij} = 0. \end{cases} \tag{1}$$

Or intuitively, at an equilibrium, each user has the same marginal value on machines where they place positive bids and has lower marginal values on those machines where they do not bid (recall that $Y_j = \sum_{i=1}^{n} x_{ij}$).

We next prove that the existence holds for any m-player game, as long as it is strongly competitive. One difficulty that prevents us from applying standard argument, e.g. Rosen's theorem [24], is the discontinuity of the game at the point where for some j, $x_{ij} = 0$ for each i. To overcome this difficulty, we consider a perturbed game Q^ε of Q and first show the existence of Nash equilibrium in Q^ε. We then argue that by letting $\varepsilon \to 0$, a limiting point of the Nash equilibria is a Nash equilibrium of the original game Q. This approach is similar to that is in [11]. We note that in [23], Reny proposed the notion of better-reply security as a general approach to proving the existence of Nash equilibrium in a game with discontinuity. It is plausible that our result can be achieved by taking that route too.

Theorem 1 *In any m-player strongly competitive game, there exists a pure Nash equilibrium.*

Proof: Recall that w_{ij} is the weight of user i on machine j, where $1 \le i \le m$ and $1 \le j \le n$. We consider strongly competitive game in which for any $1 \le j \le n$, there exist $i_1 \ne i_2$, such that $w_{i_1 j}, w_{i_2 j} > 0$. Suppose that x_{ij} is the bid of player i on machine j. Let $Y_j = \sum_{i=1}^m x_{ij}$, the total amount bid on machine j, and $z_{ij} = Y_j - x_{ij}$. Let $X = \sum_{i=1}^m X_i$, the total money in the system, and $Z_i = X - X_i$.

As mentioned above, we will show the existence of Nash equilibrim as a limit point of the Nash equilibria of a sequence of perturbed games Q^ε. Consider the purturbed game Q^ε in which each player's payoff function is defined as:

$$U_i^\varepsilon(x) = \sum_{j=1}^n w_{ij} \frac{x_{ij}}{\varepsilon + Y_j} .$$

Or we can think that there is an additional player who bids a tiny amount of money on each machine so the function U_i^ε is continuous and concave everywhere. It is easily verified that U_i^ε is concave in x_{ij}'s, for $1 \le j \le n$. The domain of player i's strategy is the set

$$\Omega_i = \{(x_{i1}, \ldots, x_{in}) \mid \sum_{j=1}^n x_{ij} = X_i \,, \quad x_{ij} \ge 0\} \,,$$

which is clearly a compact convex set. Therefore, by Rosen's theorem [24], there exists a Nash equilibrium of the game Q^ε. Pick $\omega^\varepsilon = (x_{ij}^\varepsilon)$ be any equilibrium. Let $\varepsilon \to 0$. Again, since the strategy space is compact, there exist an infinite sequence that converge to a limit point. Suppose the limit point is ω, i.e. we have a sequence $\varepsilon_k \to 0$ and $\omega^{\varepsilon_k} \to \omega$. Clearly, ω is a legitimate strategy. We shall show that ω is a Nash equilibrium of the original game $Q = Q^0$. We can safely assume that for each player i, it has non-zero weights on at least two machines. The other cases can be easily handled — if the weights in a player's utility function is all 0, we can split his money evenly on all the machines; if the utility function of a player has only one positive weight, the player will have to bid all of his money on that machine. In both cases, it will only increase the reservation price on some machines and not cause problem to our argument.

Let us consider only those ε's in the converging sequence. In what follows, a constant means a number that is solely determined by the system parameters, m, n, w_{ij}'s, and X_i's, and is independent of ε. Similarly, let $Y_j^\varepsilon = \sum_{i=1}^m x_{ij}^\varepsilon$, and $z_{ij}^\varepsilon = Y_j^\varepsilon - x_{ij}^\varepsilon$.

Lemma 1 *There exists a constant $M_0, M_1 > 0$ such that for sufficiently small ε, $M_0 \le \lambda_i^\varepsilon \le M_1$ for any $1 \le i \le m$.*

Proof: For any player i, it has to bid at least $\frac{X_i}{n}$ on some machine with positive weight, suppose it is machine j. Then,

$$\lambda_i^\varepsilon = w_{ij}\frac{\varepsilon + z_{ij}}{(\varepsilon + z_{ij} + x_{ij})^2}\,.$$

Therefore, λ_i^ε is minimized when $z_{ij} = X - X_i$ and maximized when $\varepsilon + z_{ij} = x_{ij}$. Thus,

$$\lambda_i^\varepsilon \ge w_{ij}\frac{\varepsilon + X - X_i}{(\varepsilon + X)^2}\,,$$

and

$$\lambda_i^\varepsilon \le \frac{w_{ij}}{4x_{ij}} \le \frac{nw_{ij}}{4X_i}\,.$$

Set $M_0 = \min_{w_{ij}>0} w_{ij}\frac{X-X_i}{4X^2}$ and $M_1 = \max_{w_{ij}>0} \frac{nw_{ij}}{4X_i}$. It is easy to verify that when $\varepsilon \le X$, $M_0 \le \lambda_i^\varepsilon \le M_1$. □

Lemma 2 *There exists a constant $c_0 > 0$ such that for sufficiently small ε and for any j, $Y_j^\varepsilon \ge c_0$.*

Proof: We first show that for sufficiently small ε, in the Nash equilibrium of Q^ε, there are at least two players bidding on each machine. For machine j, let w_j, W_j be, respectively, the minimum and maximum nonzero weight on j. When $\varepsilon < w_j/M_1$ (defined in Lemma 1), there must exist some player bidding on machine j because otherwise for a player i with $w_{ij} > 0$, i's margin on machine j would be $w_{ij}/\varepsilon > M_1$, contradicting with Lemma 1. Thus, there must be some player bidding on player j for sufficiently small ε. Now consider the situation when there is only one player bidding on machine j. Suppose it is player i. Then, $\lambda_i = w_{ij}\frac{\varepsilon}{(\varepsilon + x_{ij}^\varepsilon)^2}$. Since $\lambda_i \ge M_0$, $x_{ij}^\varepsilon \le \sqrt{\frac{w_{ij}\varepsilon}{M_0}} - \varepsilon$.

For another player k with nonzero weight on j (we know there must exist one by assumption), its margin λ on machine j is

$$\frac{w_{kj}}{\varepsilon + x_{ij}^{\varepsilon}} \geq w_{kj}\sqrt{\frac{M_0}{w_{ij}\varepsilon}} .$$

Therefore, when $\varepsilon < \frac{w_j^2 M_0}{W_j M_1^2}$, there must be at least two players bidding on j.

Now, suppose that there are $k \geq 2$ players bidding on machine j in Q^{ε}. Let them be player 1 to k. Clearly, all of those players have non-zero weights on j. By Lemma 1, we have that for any $1 \leq i \leq k$, $w_{ij}\frac{\varepsilon+z_{ij}^{\varepsilon}}{(\varepsilon+Y_j^{\varepsilon})^2} \leq M_1$.

Set $M_2 = \max_{w_{ij}>0} M_1/w_{ij}$. Then,

$$\frac{\varepsilon + z_{ij}^{\varepsilon}}{(\varepsilon + Y_j^{\varepsilon})^2} \leq M_1/w_{ij} \leq M_2 , \quad \text{for } 1 \leq i \leq k.$$

Thus,

$$\begin{aligned} kM_2 &\geq \sum_{i=1}^{k} \frac{\varepsilon + z_{ij}^{\varepsilon}}{(\varepsilon + Y_j^{\varepsilon})^2} = \frac{k\varepsilon + \sum_{i=1}^{k} z_{ij}^{\varepsilon}}{(\varepsilon + Y_j^{\varepsilon})^2} \\ &= \frac{k\varepsilon + (k-1)Y_j^{\varepsilon}}{(\varepsilon + Y_j^{\varepsilon})^2} > \frac{k-1}{\varepsilon + Y_j^{\varepsilon}} . \end{aligned}$$

Therefore $Y_j^{\varepsilon} > \frac{k-1}{kM_2} - \varepsilon$ for $k \geq 2$. Set $c_0 = \frac{1}{4M_2}$. When ε is sufficiently small, say $\leq \frac{1}{4M_2}$, we have that $Y_j^{\varepsilon} \geq c_0$. □

We are now ready for the main lemma,

Lemma 3 *For any $\delta > 0$, for sufficiently small ε, we have that*

$$\left| \frac{\partial U_i(x)}{\partial x_{ij}}(\omega) - \frac{\partial U_i^{\varepsilon}(x)}{\partial x_{ij}}(\omega^{\varepsilon}) \right| \leq \delta .$$

Proof:

$$\begin{aligned} \frac{\partial U_i(x)}{\partial x_{ij}}(\omega) &= w_{ij}\frac{z_{ij}}{Y_j^2} , \\ \frac{\partial U_i^{\varepsilon}(x)}{\partial x_{ij}}(\omega^{\varepsilon}) &= w_{ij}\frac{z_{ij}^{\varepsilon}}{(\varepsilon + Y_j^{\varepsilon})^2} . \end{aligned}$$

The lemma follows immediately by $z_{ij}^{\varepsilon} \to z_{ij}$ and $Y_j^{\varepsilon} \to Y_j$, and that $Y_j^{\varepsilon} \geq c_0$, for some constant $c_0 > 0$. □

Now, we are ready to show that ω is a Nash equilibrium of the game Q. Suppose it is not true, then the optimum condition is violated for some player i. There are two possibilities.

1. There are j, k, where $j \neq k$, such that $x_{ij}, x_{ik} > 0$ and $\frac{\partial U_i}{\partial x_{ij}} \neq \frac{\partial U_i}{\partial x_{ik}}$. By Lemma 3, we know that for sufficiently small ε, the following holds: $x_{ij}^{\varepsilon} > 0, x_{ik}^{\varepsilon} > 0$, and $\frac{\partial U_i^{\varepsilon}}{\partial x_{ij}}(\omega^{\varepsilon}) \neq \frac{\partial U_i^{\varepsilon}}{\partial x_{ik}}(\omega^{\varepsilon})$. It contradicts with ω^{ε} being a Nash equilibrium of Q^{ε}. Write $\lambda_i = w_{ij} \frac{\partial U_i}{\partial x_{ij}}(\omega)$ for $x_{ij} > 0$.

2. There is j such that $x_{ij} = 0$, and $w_{ij} \frac{\partial U_i}{\partial x_{ij}}(\omega) > \lambda_i$. By the same argument in 1, we can derive contradiction.

Hence, ω is a Nash equilibrium of the game Q. □

4 Efficiency

The efficiency of a game Q is defined as: $\pi(Q) = \min_{\omega \in \Omega_0} \pi(\omega)$, which is the inverse of the price of anarchy (POA). It is easy to verify that the social optimum is achieved as follows.

Claim 1 *The optimal social welfare is achieved when each machine is fully allocated to the player who has the maximum weight on the machine (where ties are broken arbitrarily). i.e.* $U^* = \sum_{j=1}^{n} \max_{1 \leq i \leq m} w_{ij}$.

If we allow arbitrary budgets and weights, then clearly the POA of the game is not bounded. For the analysis purpose, we consider only balanced games in which all the players have the same budget, 1, and for any i, $\sum_j w_{ij} = 1$.

4.1 Two-Player Games

We next show that even in the simplest nontrivial case when there are two users and two machines, the game has interesting properties. We start with two special cases to provide some intuition about the game. In what follows, we assume that each user has a bounded budget of 1. The weight matrices are shown in figure 1(a) and (b), which correspond respectively to the equal-weight and opposite-weight games. Let x and y denote the respective bids of users 1 and 2 on machine 1 (clearly, their respective bids on machine 2 are $1 - x$ and $1 - y$). Also denote $s = x + y$ and $\delta = (2 - s)/s$.

Example 1 ***Equal-weight game.*** *In Figure 1(a), both users have equal valuations for the two machines. By the optimality condition, for the bid vectors to be in equilibrium, they need to satisfy*

	m_1	m_2
u_1	α	$1-\alpha$
u_2	α	$1-\alpha$

(a) equal weight game

	m_1	m_2
u_1	α	$1-\alpha$
u_2	$1-\alpha$	α

(b) opposite weight game

Figure 1: Two special cases of two-player games.

the following equations (according to (1)):

$$\begin{aligned} \alpha\frac{y}{(x+y)^2} &= (1-\alpha)\frac{1-y}{(2-x-y)^2} \\ \alpha\frac{x}{(x+y)^2} &= (1-\alpha)\frac{1-x}{(2-x-y)^2} \end{aligned}$$

By simplifying the above equations, we obtain that $\delta = 1 - 1/\alpha$ *and* $x = y = \alpha$. *Thus, there exists a unique Nash equilibrium of the game where the two users have the same bidding vector. At the equilibrium, the utility of each user is* $1/2$, *and the social welfare is* 1. *Since OPT=1 as well,* $\pi(Q) = 1$.

Example 2 ***Opposite-weight game.*** *The situation is different for the opposite game in which the two users put the exact opposite weights on the two machines. Assume WLOG that* $\alpha \geq 1/2$. *Similarly, for the bid vectors to be at the equilibrium, they need to satisfy*

$$\begin{aligned} \alpha\frac{y}{(x+y)^2} &= (1-\alpha)\frac{1-y}{(2-x-y)^2} \\ (1-\alpha)\frac{x}{(x+y)^2} &= \alpha\frac{1-x}{(2-x-y)^2} \end{aligned}$$

By simplifying the above equations, we have that each Nash equilibrium corresponds to a non-negative root of the cubic equation $f(\delta) = \delta^3 - c\delta^2 + c\delta - 1 = 0$, *where* $c = \frac{1}{2\alpha(1-\alpha)} - 1$.

Clearly, $\delta = 1$ *is a root of* $f(\delta)$. *When* $\delta = 1$, *we have that* $x = \alpha$, $y = 1 - \alpha$, *which is the symmetric equilibrium that is consistent with our intuition — each user puts a bid proportional to his preference of the machine. At this equilibrium,* $U = 2 - 4\alpha(1-\alpha)$, $U^* = 2\alpha$, *and* $U/U^* = (2\alpha + \frac{1}{\alpha}) - 2$, *which is minimized when* $\alpha = \frac{\sqrt{2}}{2}$ *with the minimum value of* $2\sqrt{2} - 2 \approx 0.828$. *However, when* α *is large enough, there exist two other roots, corresponding to less intuitive asymmetric equilibria.*

Intuitively, the asymmetric equilibrium arises when user 1 *values machine* 1 *a lot, but by placing even a relatively small bid on machine* 1, *he can get most of the machine because user* 2 *values* *machine* 1 *very little, and thus places an even smaller bid. In this case, user* 1 *gets most of machine* 1 *and almost half of machine* 2.

The threshold is achieved when $f'(1) = 0$, *i.e. when* $c = \frac{1}{2\alpha(1-\alpha)} - 1 = 3$. *This solves to* $\alpha_0 = \frac{2+\sqrt{2}}{4} \approx 0.854$. *Those asymmetric equilibria at* $\delta \neq 1$ *are "bad" as they yield lower efficiency than the symmetric equilibrium. Let* δ_0 *be the minimum root. When* $\alpha \to 0$, $c \to +\infty$, *and* $\delta_0 = 1/c + o(1/c) \to 0$. *Then,* $x, y \to 1$. *Thus,* $U \to 3/2$, $U^* \to 2$, *and* $U/U^* \to 0.75$.

From the above simple game, we already observe that the Nash equilibrium may not be unique, which is different from many congestion games in which the Nash equilibrium is unique.

The following theorem shows that the above efficiency is in fact the worst case.

Theorem 2 *For a two player game,* $\pi(Q) \geq 3/4$, *and the bound is tight.*

The proof of above theorem follows from the following two lemmas, whose proofs are included in the appendix.

Lemma 4 *The lowest efficiency can be realized by a two machine balanced game.*

Lemma 5 *When there are two machines and two players, the efficiency is at least* $3/4$.

4.2 m-Player Games

We next present efficiency bounds for the m-player game.

Theorem 3 *For the m-player game* Q, $\pi(Q) = \Omega(1/\sqrt{m})$, *and the bound is tight in the worst case.*

In what follows, we provide an example demonstrating the lower bound. For the upper bound proof, refer to [33].

Example 3 *Consider a game with* $m = n^2 + n$ *players and* n *machines. Of the players, there are* n^2 *who have the same weights on all the machines, i.e.* $1/n$ *on each machine. The other* n *players have weight* 1, *each on a different machine and* 0 *(or a sufficiently small* ϵ*) on all the other machines. Clearly,* $U^* = n$. *The following allocation is an equilibrium: the first* n^2 *players evenly distribute their money among all the machines, the other* n *player bid all of their money on their respective favorite machine. Hence, the total money on each machine is* $n + 1$. *At this equilibrium,* *each of the first n^2 players receives $\frac{1}{n}\frac{1/n}{n+1} = \frac{1}{n^2(n+1)}$ on each machine, resulting in a total utility of $n^3 \cdot \frac{1}{n^2(n+1)} < 1$. The other n players each receives $\frac{1}{n+1}$ on their favorite machine, resulting in a total utility of $n \cdot \frac{1}{n+1} < 1$. Therefore, the total utility of the equilibrium is < 2, while the social optimum is $n = \Theta(\sqrt{m})$. This bound is the worst possible.*

5 Fairness

To measure the fairness of the equilibrium, we use two metrics, namely, utility uniformity and envy freeness.

5.1 Utility Uniformity

The utility uniformity is defined as the ratio between the highest and the lowest utility levels that a user achieves in the game. We begin with an analysis of the two simple games presented in examples 1 and 2 in Section 4. In the equal-weight game, both users receive the same utility at the unique equilibrium, therefore $\tau(Q) = 1$. In contrast, in the opposite-weight game, when α approaches 1, there exists an asymmetric equilibrium in which the utility uniformity approaches 1/2. We next show that this is the worst case for a two-player game.

Theorem 4 *For the m-player game Q, $\tau(Q) \geq 1/m$, and the bound is tight in the worst case.*

Proof: Let $(S_1, \ldots, S_n)$ be the current total bids on the n machines, excluding user i. User i can ensure a utility of $1/m$ by distributing his budget proportionally to the current bids. That is, user i, by bidding $s_{ij} = S_i / \sum_{i=1}^n S_i$ on machine j, obtains a resource level of:

$$r_{ij} = \frac{s_{ij}}{s_{ij} + S_j} = \frac{S_j / \sum_{i=1}^n S_i}{S_j / \sum_{i=1}^n S_i + S_j} = \frac{1}{1 + \sum_{i=1}^n S_i},$$

where $\sum_{j=1}^n S_j = \sum_{j=1}^m X_j - X_i = m - 1$.

Therefore, $r_{ij} = \frac{1}{1+m-1} = \frac{1}{m}$. The total utility of user i is

$$\sum_{j=1}^{n} r_{ij} w_{ij} = (1/m) \sum_{j=1}^{n} w_{ij} = 1/m.$$

Since each user's utility cannot exceed 1, the minimal possible uniformity is $1/m$.

To see that the bound is tight, we consider the case when $w_{11} = 1$, $w_{1j} = 0$ for $2 \leq j \leq n$, $w_{i1} = \epsilon$ for $2 \leq i \leq m$, and $w_{ij} = (1-\epsilon)/(n-1)$ for $2 \leq i \leq m$ and $2 \leq j \leq n$. It is easily verified that for sufficiently small ϵ, $U_1 = 1$ and $U_i \approx 1/m$ for $2 \leq i \leq m$. □

5.2 Envy-Freeness

We begin our discussion with the simple 2-player games presented in section 4. It is easy to verify that both games are envy free. More generally, we have that:

Claim 2 *Any two player game is envy free.*

Proof: Suppose that the two user's shares are $\mathbf{r}_1 = (r_{11}, \ldots, r_{1n})$ and $\mathbf{r}_2 = (r_{21}, \ldots, r_{2n})$ respectively. Then $U_1(\mathbf{r}_1) + U_1(\mathbf{r}_2) = U_1(\mathbf{r}_1 + \mathbf{r}_2) = U_1(1, \ldots, 1) = 1$ because $r_{i1} + r_{i2} = 1$ for all $1 \leq i \leq n$. Again by that $U_1(\mathbf{r}_1) \geq 1/2$ (Theorem 4), we have that $U_1(\mathbf{r}_1) \geq U_1(\mathbf{r}_2)$, i.e. any equilibrium allocation is envy-free. $\square$

For m-player games, while the utility uniformity can be small, the envy-freeness, on the other hand, is bounded by a constant of $2\sqrt{2} - 2 \approx 0.828$, as shown in [33].

Theorem 5 *For the m-player game Q, $\rho(Q) \geq 2\sqrt{2} - 2$, and the bound is tight in the worst case.*

6 Algorithms

In the previous sections, we proved analytically the existence of a Nash equilibrium in the bounded-budget resource allocation game. We also provided upper and lower bounds on the efficiency and fairness of the allocation game. Yet, it is unclear whether and under what strategies the game will converge to an equilibrium. In particular, we would like to find out whether the intuitive strategy of each player constantly re-adjusting his bids according to the best-response algorithm leads to an equilibrium. In addition, it is interesting to examine the expected efficiency and fairness under reasonable distributions of the players' preferences. To answer these questions, we resort to simulations.

In this section, we present the algorithms that we use to compute the social optimum and the (approximated) best response in our experiments. We consider both the infinite parallelism and finite parallelism models.

6.1 Infinite Parallelism Model

6.1.1 Best-Response

We now present the algorithm for computing the best response in the infinite-parallelism model. Recall that for weights $w_1, \ldots, w_n$, total bids $y_1, \ldots, y_n$, and the budget X, the best response is to solve the following optimization problem

$$\text{maximize } U = \sum_{j=1}^{n} w_j \tfrac{x_j}{x_j + y_j} \text{ subject to}$$

$$\sum_{j=1}^{n} x_j = X, \text{ and } x_j \geq 0.$$

To compute the best response, we first sort $\frac{w_j}{y_j}$ in decreasing order. Without loss of generality, suppose that

$$\frac{w_1}{y_1} \geq \frac{w_2}{y_2} \geq \ldots \frac{w_n}{y_n} \,.$$

Suppose that $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ is the optimal solution.

Claim 3 *If $x_i^* = 0$, then it must hold that for any $j > i$, $x_j^* = 0$.*

Proof: Suppose this were not true. Then

$$\begin{aligned} \frac{\partial U}{\partial x_j}(\mathbf{x}^*) &= w_j \frac{y_j}{(x_j^* + y_j)^2} < w_j \frac{y_j}{y_j^2} \\ &= \frac{w_j}{y_j} \leq \frac{w_i}{y_i} = \frac{\partial U}{\partial x_i}(\mathbf{x}^*) \,. \end{aligned}$$

Thus it contradicts with the optimality condition (1). □

Suppose that $k = \max\{i | x_i^* > 0\}$. Again, by the optimality condition, there exists λ such that $w_i \frac{y_i}{(x_i^* + y_i)^2} = \lambda$ for $1 \leq i \leq k$, and $x_i^* = 0$ for $i > k$. Equivalently, we have that:

$$x_i^* = \sqrt{\frac{w_i y_i}{\lambda}} - y_i \,, \text{for } 1 \leq i \leq k, \text{ and } x_i^* = 0 \text{ for } i > k.$$

Replacing them in the equation $\sum_{i=1}^{n} x_i^* = X$, we can solve for $\lambda = \frac{(\sum_{i=1}^{k} \sqrt{w_i y_i})^2}{(X + \sum_{i=1}^{k} y_i)^2}$. Thus,

$$x_i^* = \frac{\sqrt{w_i y_i}}{\sum_{i=1}^{k} \sqrt{w_i y_i}} (X + \sum_{i=1}^{k} y_i) - y_i \,.$$

The remaining question is how to determine k. It is the largest value such that $x_k^* > 0$. Thus, we obtain the following algorithm to compute the best response of a user:

1. Sort the machines according to $\frac{w_i}{y_i}$ in decreasing order.

2. Compute the largest k such that

$$\frac{\sqrt{w_k y_k}}{\sum_{i=1}^{k} \sqrt{w_i y_i}}(X + \sum_{i=1}^{k} y_i) - y_k \geq 0.$$

3. Set $x_j = 0$ for $j > k$, and for $1 \leq j \leq k$, set:

$$x_j = \frac{\sqrt{w_j y_j}}{\sum_{i=1}^{k} \sqrt{w_i y_i}}(X + \sum_{i=1}^{k} y_i) - y_j.$$

The computational complexity of this algorithm is $O(n \log n)$, dominated by the sorting. In practice, the best response can be computed infrequently (e.g. once a minute), so for a typically powerful modern host, this cost is negligible.

The best response algorithm must send and receive $O(n)$ messages because each user must obtain the total bids from each host. In practice, this is more significant than the computational cost. Note that hosts only reveal to users the sum of the bids placed on them. As a result, hosts do not reveal the private preferences and even the individual bids of one user to another.

We next demonstrate that best-response dynamics do not guarantee convergence to a NE.

Claim 4 *The best-response dynamics (where players re-adjusting their bids based on the best-response algorithm) do not necessarily converge to a NE of the game.*

Proof: Consider the example with four players and two machines. Let $\varepsilon, \delta > 0$ be small constant. Let the weights be

$$\begin{aligned} w_{11} &= \varepsilon, & w_{12} &= 1 - \varepsilon; \\ w_{21} &= 1 - \varepsilon, & w_{22} &= \varepsilon; \\ w_{31} &= 1, & w_{32} &= 0; \\ w_{41} &= 0, & w_{42} &= 1. \end{aligned}$$

Let the players' budgets be $X_1 = 1$, $X_2 = 1$, $X_3 = \delta$, and $X_4 = \delta$. In this example, player 3 and 4 always place all of their bids on machine 1 and 2, respectively. It is easily verified that when player 1 and 2 start with an even split of bids, the best response, alternatively performed by player 1 and 2, cycles. $\square$

Despite this example, the simulations in the next section show that large oscillations should not be expected under the distribution of preferences we consider. The question of whether there exists a distribution of preferences that guarantees convergence of the best-response dynamics to a NE remains an open question.

Open question 1 *Are there distribution of preferences that guarantee convergence of the best-response dynamics to a NE?*

6.2 Finite Parallelism Model

In practice, some applications may contain inherent limits on parallelization. To model this, we consider the finite parallelism model, where each user i only places bids on at most k_i machines. Obviously, the infinite parallelism model is a special case of the finite parallelism model in which $k_i = n$ for all the i's. In the finite parallelism model, computing the social optimum is no longer trivial due to the bounded parallelism. Yet, it can be computed by using the maximum matching algorithm.

Consider the weighted complete bipartite graph $G = U \times V$, where $U = \{u_{i\ell} | 1 \leq i \leq m \text{ , and } 1 \leq \ell \leq k_i\}$, $V = \{1, 2, \ldots, n\}$ with edge weight w_{ij} assigned to the edge $(u_{i\ell}, v_j)$. A matching of G is a set of edges with disjoint nodes, and the weight of a matching is the total weights of the edges in the matching. As a result, the following lemma holds.

Lemma 6 *The social optimum in the finite-parallelism model is obtained as the maximum weight matching of G.*

While the maximum-weight matching is a classical network problem that can be solved in polynomial time, for example by the Hungarian algorithm [15] in $O((\sum_i k_i + n)^3)$ time, we do not know an efficient algorithm to compute the best response under the finite parallelism model. Instead, we provide the following local search heuristic.

Suppose we have n machines with weights $w_1, \ldots, w_n$ and total bids $y_1, \ldots, y_n$. Let the user's budget be X and the parallelism bound be k. Our goal is to compute an allocation of X to up to k machines to maximize the user's utility.

For a subset of machines A, denote by $\mathbf{x}(A)$ the best response on A without parallelism bound and by $U(A)$ the utility obtained by the best response algorithm. The local search works as follows:

1. Set A to be the k machines with the highest w_i/y_i.
2. Compute $U(A)$ by the infinite parallelism best response algorithm (Sec 6.1) on A.
3. For each $i \in A$ and each $j \notin A$, repeat
4. Let $B = A - \{i\} + \{j\}$, compute $U(B)$.
5. If($U(B) > U(A)$), let $A \leftarrow B$, and goto 2.
6. Output $\mathbf{x}(A)$.

Intuitively, by the local search heuristic, we test if we can swap a machine in A for one not in A to improve the best response utility. Clearly, this can be done efficiently by the best response algorithm in the infinite parallelism model. If so, we swap the machines and repeat the process. Otherwise, we have reached a local maxima and output that value. We suspect, supported by the experimental evidence, that the local maxima that this algorithm finds is also the global maximum and that this process stops after a polynomial number of iterations, but we are unable to establish it. See section 7 for the dynamics obtained from this algorithm.

6.3 Local Greedy Adjustment

The algorithms discussed above are specific to the linear utility function assumed in our model. In practice, utility functions may have a more complicated form, or even worse, a user may not have a formulation of his utility function. We do assume that the user still has a way to measure his utility, which is the minimum assumption necessary for any market-based resource allocation mechanism. In these situations, users can use a more general strategy, the local greedy adjustment method, which works as follows. A user finds the two machines that provide him with the highest and lowest marginal utilities. He then moves a fixed small amount of money from the machine with the lowest marginal utility to the highest one. This strategy aims to adjust the bids so that the marginal values of each machine being bid on are the same. This is a sufficient condition for an optimal allocation for concave utility functions. The tradeoff here is the low convergence rate.

7 Simulation Results

While the analytic results provide us with worst-case analysis for the infinite parallelism model, in this section we employ simulations to study the properties of the Nash equilibria in more realistic scenarios and for the finite parallelism model. First, we determine whether the user bidding process converges, and if so, what the rate of convergence is. Second, in cases of convergence, we look at the performance at equilibrium, using the efficiency and fairness metrics defined above.

Iterative Method. In our simulations, each user starts with an initial bid vector and then iteratively updates his bids until a convergence criterion (described below) is met. The initial bid is set proportional to the user's weights on the machines. We experiment with two update methods, the best response methods, as described in Section 6.1 and 6.2, and the local greedy adjustment method, as described in Section 6.3.

Convergence Criteria. Convergence time measures how quickly the system reaches an equilibrium. It is particularly important in the highly dynamic environment of distributed shared clusters, in which the system's conditions may change before reaching the equilibrium. In some cases, a high convergence rate may be more significant than the efficiency at the equilibrium.

By definition, the system converges to a Nash equilibrium if the best-response bids of each user at time t is equal to his bids at time $t-1$. Yet, in our simulations, we require a weaker condition. We believe that users may not be willing to re-allocate their bids dramatically for a small utility gain. Therefore, we use the *utility gap* criterion for convergence. We say that the system has converged if the utility gap of each user is smaller than ϵ (0.001 in our experiments).

The utility gap criterion does not fit the local greedy adjustment method, because under this method users will experience constant fluctuations in utility as they move money around. For this method, we use the *marginal utility gap* criterion. We compare the highest and lowest utility margins over all the machines. According to this criterion, the system converges if the marginal utility gap of each user is negligible.

A related issue is the stabilization of the social welfare in the system. This might be interesting from the system provider's point of view. We call it the *social welfare stabilization* criterion. We say that the system has stabilized if the change in social welfare is $\leq \epsilon$. Note that it is possible that the system will stabilize under this criterion, while individual users' utilities may not have converged. This criterion is useful to evaluate how quickly the system as a whole reaches a particular efficiency level.

User preferences. We experiment with two models of user preferences:

- uniform distribution: users' weights on the different machines are independently and identically distributed, according the uniform distribution.

- correlated distribution: in practice, users' preferences may be correlated based on factors like the hosts' location and the types of applications that users run. To capture these correlations, we associate with each user and machine a resource profile vector where each dimension of the vector represents one resource (e.g., CPU, memory, and network bandwidth). For a user i with a profile $\mathbf{p}_i = (p_{i1}, \ldots, p_{i\ell})$, p_{ik} represents user i's need for resource k. For machine j with profile $\mathbf{q}_j = (q_{j1}, \ldots, q_{j\ell})$, q_{jk} represents machine j's strength with respect to resource k. Then, w_{ij} is the dot product of user i's and machine j's resource profiles, i.e. $w_{ij} = \mathbf{p}_i \cdot \mathbf{q}_j = \sum_{k=1}^{\ell} p_{ik} q_{jk}$. By using these profiles, we compress the parameter space and introduce correlations between users and machines.

In the following simulations, we fix the number of machines to 100 and vary the number of users from 5 to 250 (we only report the results for the range of $5 - 150$ users since the results remain similar for a larger number of users). Sections 7.1 and 7.2 present the simulation results when we apply the infinite parallelism and finite parallelism models, respectively. If the system converges, we report the number of iterations until convergence. A convergence time of 200 iterations indicates non-convergence, in which case we report the efficiency and fairness values at the point we terminate the simulation.

7.1 Infinite parallelism

In this section, we apply the infinite parallelism model, which assumes that users can use an unlimited number of machines. We present the efficiency and fairness at the equilibrium. Our benchmark allocations are the social optimum allocation (allocating each machine to the user who likes it the most), and the weight-proportional allocation. In the weight-proportional allocation, each user distributes his bids proportionally to his weights on the machines.

We present results for the two user preference models. With uniform preferences, users' weights for the different machines are independently and identically distributed according to the uniform distribution, $U \sim (0,1)$ (and are normalized thereafter). In correlated preferences, each user's and each machine's resource profile vector has three dimensions, and their values are also taken from the uniform distribution, $U \sim (0,1)$.

Convergence Time. Figure 2 shows the convergence time, efficiency and fairness of the infinite parallelism model under uniform (left) and correlated (right) preferences. Plots (a) and (b) show the convergence and stabilization time of the best-response and local greedy adjustment methods. The best-response algorithm converges within a few number of iterations for any number of users. In contrast, the local greedy adjustment algorithm does not converge even within 500 iterations when the number of users is smaller than 60, but does converge for a larger number of users. We believe that for small numbers of users, there are dependency cycles among the users that prevent the system from converging because one user's decisions affects another user, whose decisions affect another user, etc. Regardless, the local greedy adjustment method stabilizes (in terms of social welfare) within 100 iterations.

Figure 3(a) presents the efficiency over time for a system with 40 users. It demonstrates that while both adjustment methods reach the same social welfare, the best-response algorithm is significantly faster.

In the remainder of this paper, we will refer to the (Nash) equilibrium, independent of the adjustment method used to reach it.

Efficiency. Figure 2 (c) and (d) present the efficiency as a function of the number of users. We present the efficiency at equilibrium, and use the social optimum and the weight-proportional static allocation methods for comparison. Social optimum provides an efficient allocation by definition. For both user preference models, the efficiency at the equilibrium is approximately 0.9, independent of the number of users, which is only slightly worse than the social optimum. The efficiency at the equilibrium is $\approx 50\%$ improvement over the weight-proportional allocation method for uniform preferences, and $\approx 30\%$ improvement for correlated preferences.

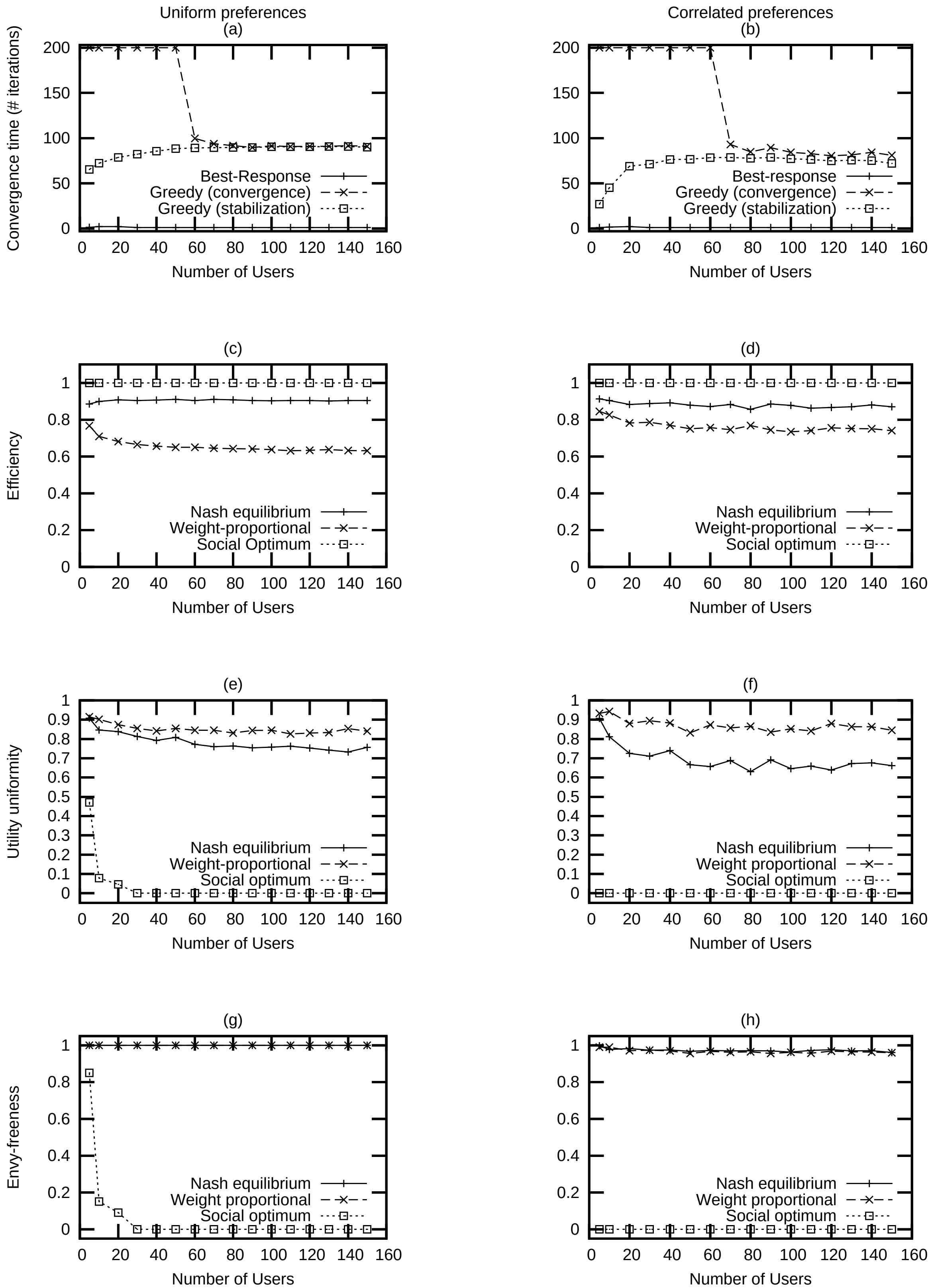
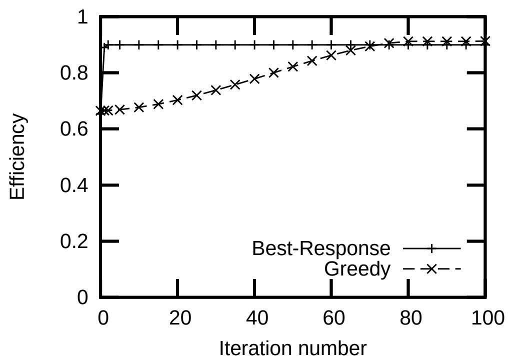
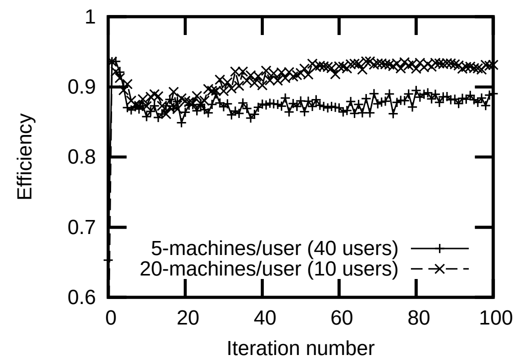


Figure 2: Efficiency, utility uniformity, enviness and convergence time as a function of the number of users under the infinite parallelism model, with uniform and correlated preferences. $n = 100$.



(a)



(b)

Figure 3: (a) Efficiency level over time under the infinite parallelism model. number of users = 40. $n = 100$. (b) Efficiency level over time under the finite parallelism model with local search algorithm. $n = 100$.

Fairness. Figure 2(e) and (f) present the utility uniformity as a function of the number of users, and figures (g) and (h) present the envy-freeness. While the social optimum yields perfect efficiency, it has poor fairness. The weight-proportional method achieves the highest fairness among the three allocation methods, but the fairness at the equilibrium is close.

The utility uniformity is slightly better at the equilibrium under uniform preferences (> 0.7) than under correlated preferences (> 0.6), since when users' preferences are more aligned, users' happiness is more likely going to be at the expense of each other. Although utility uniformity decreases in the number of users, it remains reasonable even for a large number of users, and flattens out at some point. At the social optimum, utility uniformity can be infinitely poor, as some users may be allocated no resources at all. The same is true with respect to envy-freeness. The difference between uniform and correlated preferences is best demonstrated in the social optimum results. When the number of users is small, it may be possible to satisfy all users to some extent if their preferences are not aligned, but if they are aligned, even with a very small number of users, some users get no resources, thus both utility uniformity and envy-freeness go to zero. As the number of users increases, it becomes almost impossible to satisfy all users independent of the existence of correlation.

These results demonstrate the tradeoff between the different allocation methods. The efficiency at the equilibrium is lower than the social optimum, but it performs much better with respect to fairness. The equilibrium allocation is completely envy-free under uniform preferences and almost envy-free under correlated preferences.

7.2 Finite parallelism

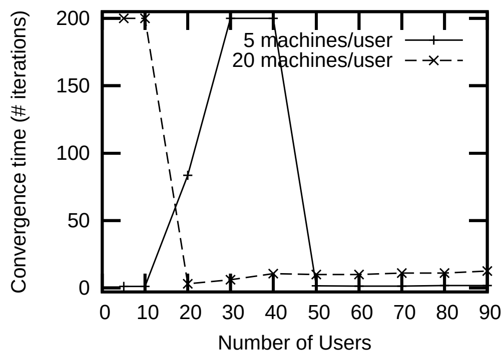


Figure 4: Convergence time under the finite parallelism model. $n = 100$.

For the finite-parallelism model, we use the local search algorithm, as described in Section 6.2, to adjust user's bids. We again experimented with both the uniform and correlated preferences distributions and did not find significant differences in the results so we present the simulation results for only the uniform distribution.

In our experiments, the local search algorithm stops quickly — it usually discovers a local maximum within two iterations. As mentioned before, we cannot prove that a local maximum is the global maximum, but our experiments indicate that the local search heuristic leads to high efficiency level.

Convergence time. Let Δ denote the parallelism bound that limits the maximum number of machines each user can bid on. We experiment with $\Delta = 5$ and $\Delta = 20$. In both cases, we use 100 machines and vary the number of users. Figure 4 shows that the system does not always converge, but if it does, the convergence happens quickly. The non-convergence occurs when the number of users is between 20 and 40 for $\Delta = 5$, between 5 and 10 for $\Delta = 20$. We believe that the non-convergence is caused by moderate competition. No competition allows the system to equilibrate quickly because users do not have to change their bids in reaction to changes in others' bids. High competition also allows convergence because each user's decision has only a small impact on other users, so the system is more stable and can gradually reach convergence. However, when there is moderate competition, one user's decisions may cause dramatic changes in another's decisions and cause large fluctuations in bids. In both cases of non-convergence, the ratio of "competitors" per machine, $\delta = m \times \Delta/n$ for m users and n machines, is in the interval $[1, 2]$. Although the system does not converge in these "bad" ranges, the system nonetheless achieves and maintains a high level of overall efficiency after a few iterations (as shown in Figure 3(b)).

Performance. In Figure 5, we present the efficiency, utility uniformity, and envy-freeness at the Nash equilibrium for the finite parallelism model. When the system does not converge, we measure performance by taking the minimum value we observe after running for many iterations. When $\Delta = 5$, there is a performance drop, in particular with respect to the fairness metrics, in the range between 20 and 40 users (where it does not converge). For a larger number of users, the system converges and achieves a lower level of utility uniformity, but a high degree of efficiency and envy-freeness, similar to those under the infinite parallelism model. As described above, this is due the competition ratio falling into the "head-to-head" range. When the parallelism bound is large ($\Delta = 20$), the performance is closer to the infinite parallelism model, and we do not observe this drop in performance.

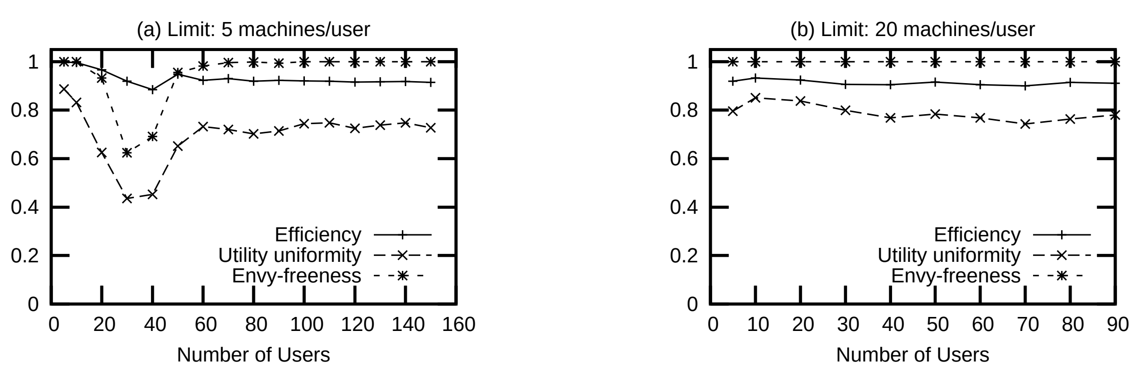


Figure 5: Efficiency, utility uniformity and envy-freeness under the finite parallelism model. $n = 100$.

8 Conclusions

This work studies the performance of a market-based proportional-share mechanism for allocating of shared resources among users with bounded budgets, using both analytical and simulation methods. We analytically derive upper and lower bounds on the efficiency and fairness of this allocation mechanism. While the analytic bounds may be poor, our simulation results show that, under some assumptions on the distribution of users' preferences, the system stabilizes at a high performance level for both efficiency and fairness. In addition, with a few exceptions under the finite parallelism model, the system reaches equilibrium quickly by using the best response algorithm and, when the number of users is not too small, by the greedy local adjustment method.

Our model and results suggest several interesting directions for future research. One is to consider more realistic utility functions. For example, we assume that there is no parallelization cost, and there is no performance degradation when multiple users share the same machine. In practice, both assumptions may not hold. For examples, in networked shared clusters, the user must copy code and data to a machine before running his application there, and there is overhead for multiplexing resources on a single machine. When the job size is large enough and the degree of multiplexing is sufficiently low, we can probably ignore those effects, but those costs should be taken into account for a more realistic modeling. Another assumption is that users have infinite work, so the more resources they can acquire, the better. In practice, users have finite work. One approach to address this is to model the user's utility according to the time to finish a task rather than the amount of resources he receives.

Another direction is to study the dynamic properties of the system when the users' needs change over time, according to some statistical model. In addition to the usual questions concerning repeated games, it would also be important to understand how users should allocate their budgets wisely over time to accomodate future needs.

9 Acknowledgments

We thank Bernardo Huberman, Lars Rasmusson, Eytan Adar, and Moshe Babaioff for fruitful discussions. We thank anonymous referees for many valuable comments.

References

[1] Tycoon. `http://tycoon.hpl.hp.com`. Retrieved October 26, 2006.

[2] A. AuYoung, B. N. Chun, A. C. Snoeren, and A. Vahdat. Resource Allocation in Federated Distributed Computing Infrastructures. In *Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT InfraStructure*, 2004.

[3] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. Operating System Support for Planetary-Scale Network Services. In *Symposium on Networked Systems Design and Implementation*, 2004.

[4] W. Brainard and H. Scarf. How to compute equilibrium prices in 1891. *Cowles Foundation Discussion Paper*, 1270, 2000.

[5] B. N. Chun and D. E. Culler. Market-based Proportional Resource Sharing for Clusters. Technical Report CSD-1092, University of California at Berkeley, Computer Science Division, January 2000.

[6] B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A. Warfield, P. Barham, and R. Neugebauer. Xen and the Art of Virtualization. In *Proceedings of the ACM Symposium on Operating Systems Principles*, 2003.

[7] E. Eisenberg and D. Gale. Consensus of subjective probabilities: the Pari-Mutuel method. *Annals of Mathematical Statistics*, 30:165–168, 1959.

[8] D. Ferguson, Y. Yemini, and C. Nikolaou. Microeconomic Algorithms for Load Balancing in Distributed Computer Systems. In *International Conference on Distributed Computer Systems*, pages 491–499, 1988.

[9] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, pages 83–92, New York, NY, USA, 1998. ACM Press.

[10] B. Hajek and S. Yang. Strategic Buyers in a Sum Bid Game for Flat Networks. Manuscript, `http://tesla.csl.uiuc.edu/~hajek/Papers/HajekYang.pdf`, 2004.

[11] R. Johari and J. N. Tsitsiklis. Efficiency Loss in a Network Resource Allocation Game. *Mathematics of Operations Research*, 2004.

[12] F. P. Kelly. Charging and Rate Control for Elastic Traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[13] F. P. Kelly and A. K. Maulloo. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Operational Research Society*, 49:237–252, 1998.

[14] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[15] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Res. Logis. Quart.*, 2:83–97, 1955.

[16] K. Lai, L. Rasmusson, E. Adar, S. Sorkin, L. Zhang, and B. A. Huberman. Tycoon: an Implemention of a Distributed Market-Based Resource Allocation System. *Multiagent and Grid Systems*, 1(3):169–182, Aug. 2005.

[17] I. Milchtaich. Congestion Games with Player-Specific Payoff Functions. *Games and Economic Behavior*, 13:111–124, 1996.

[18] D. Monderer and L. S. Shapley. Potential Games. *Games and Economic Behavior*, 14:124–143, 1996.

[19] J. Nieh, C. Vaill, and H. Zhong. Virtual-Time Round-Robin: An O(1) Proportional Share Scheduler. In *Proceedings of the USENIX Annual Technical Conference*, 2001.

[20] C. H. Papadimitriou. Algorithms, Games, and the Internet. In *Symposium on Theory of Computing*, 2001.

[21] J. Regehr. Some Guidelines for Proportional Share CPU Scheduling in General-Purpose Operating Systems. In *Work in progress session of the 22nd IEEE Real-Time Systems Symposium (RTSS 2001)*, London, UK, Dec. 2001.

[22] O. Regev and N. Nisan. The Popcorn Market: Online Markets for Computational Resources. In *Proceedings of 1st International Conference on Information and Computation Economies*, pages 148–157, 1998.

[23] P. J. Reny. On the existence of pure and mixed strategy Nash equilibria in discontinuous games. *Econometrica*, 67(5):1029–1056, 1999.

[24] J. B. Rosen. Existence and Uniqueness of Equilibrium Points for Concave N-person Games. *Econometrica*, 33(3):520–534, 1965.

[25] R. W. Rosenthal. A Class of Games Possessing Pure-Strategy Nash Equilibria. *Internation Journal of Game Theory*, 2:65–67, 1973.

[26] S. Sanghavi and B. Hajek. Optimal Allocation of a Divisible Good to Strategic Buyers. Manuscript, `http://tesla.csl.uiuc.edu/~hajek/Papers/OptDivisible.pdf`, 2004.

[27] L. Shapley and M. Shubik. Trade using one commodity as a means of payment. *Journal of Political Economy*, 85:937–968, 1977.

[28] I. Stoica, H. Abdel-Wahab, K. Jeffay, S. Baruah, J. Gehrke, and G. C. Plaxton. A Proportional Share Resource Allocation Algorithm for Real-Time, Time-Shared Systems. In *IEEE Real-Time Systems Symposium*, December 1996.

[29] H. R. Varian. Equity, Envy, and Efficiency. *Journal of Economic Theory*, 9:63–91, 1974.

[30] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A Distributed Computational Economy. *Software Engineering*, 18(2):103–117, 1992.

[31] C. A. Waldspurger and W. E. Weihl. Lottery Scheduling: Flexible Proportional-Share Resource Management. In *Operating Systems Design and Implementation*, pages 1–11, 1994.

[32] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Auction Protocols for Decentralized Scheduling. *Games and Economic Behavior*, 35:271–303, 2001.

[33] L. Zhang. The efficiency and fairness of a fixed budget resource allocation game. In *International Colloquium on Automata, Languages and Programming*, pages 485–496, 2005.

A Proof of Lemma 4

Proof: Suppose that the weights for the two users are $u_1, \ldots, u_n$ and $w_1, \ldots, w_n$, respectively. Denote by $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ the bids for the two users on the machines. We show that the worst case can always be achieved when there are only two machines.

Lemma 7 *If there is an allocation such that* $\frac{\partial U_1}{\partial x_i} \geq \lambda_1$ *and* $\frac{\partial U_2}{\partial y_i} \geq \mu_1$*, then there exist a Nash equilibrium with margin* λ, μ *where* $\lambda \geq \lambda_1$ *and* $\mu \geq \mu_1$*.*

Proof: Follows from that $\frac{\partial U_1}{\partial x_i} \geq \lambda_1$ is a convex set. We can restrict the strategy to the set Ω satisfying

$$
\begin{aligned}
u_i \frac{y_i}{(x_i + y_i)^2} &\geq \lambda_1 \\
w_i \frac{x_i}{(x_i + y_i)^2} &\geq \mu_1 \\
\sum_{i=1}^{n} x_i &= X \\
\sum_{i=1}^{n} y_i &= Y \\
x_i, y_i &\geq 0
\end{aligned}
$$

Since Ω is convex and non-empty, there exists a Nash equilibrium, say $\omega = (\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$, in the restricted strategy set Ω. We show that it is a Nash equilibrium in the full strategy space. Otherwise, suppose that $x = (x_1, \ldots, x_n)$ is not the best response to $y = (y_1, \ldots, y_n)$. Then, there are i, j such that $u_i \frac{y_i}{(x_i+y_i)^2} \neq u_j \frac{y_j}{(x_j+y_j)^2}$. Let $u_i \frac{y_i}{(x_i+y_i)^2} > u_j \frac{y_j}{(x_j+y_j)^2} \geq \lambda_0$. We let $x_i^* = x_i + \epsilon$ and $x_j^* = x_j - \epsilon$. Let x^* be the strategy with x_i replaced by x_i^* and x_j by x_j^*. For sufficiently small ϵ, we have that $U_1(x^*, y) > U_2(x, y)$ by $u_i \frac{y_i}{(x_i+y_i)^2} > u_j \frac{y_j}{(x_j+y_j)^2}$. Further, $x^* \in \Omega$, contradicting with that x is the best response to y in Ω. $\Box$

Now, we show that if we merge two machines, it will only decrease the social welfare at the Nash equilibrium. By merging two machines, i and j, we mean that we replace the two machines with a new machine with weights $u_i + u_j$ and $w_i + w_j$ for the two users, respectively.

Lemma 8 *Let Q_1 be obtained from Q by merging any two machines, then we have that $U(Q_1) \leq U(Q)$.*

Proof: Let $\omega = (x_1, \ldots, x_n, y_1, \ldots, y_n)$ be a Nash equilibrium of Q. Then,

$$\begin{aligned} U_1(\omega) &= \sum_{j=1}^{n} u_i \frac{x_i}{x_i + y_i} = \sum_{j=1}^{n} u_i - u_i \frac{y_i}{x_i + y_i} \\ &= \sum_{j=1}^{n} u_i - \sum_{j=1}^{n} \lambda(x_i + y_i) = 1 - 2\lambda\,. \end{aligned}$$

Similarly, $U_2(\omega) = 1 - 2\mu$.

Now, we show a stronger statement that for any Nash equilibrium ω of Q, there exists a Nash equilibrium ω_1 of Q_1 such that

$$U_i(Q_1, \omega_1) \leq U_i(Q, \omega)\,, \quad \text{for any } 1 \leq i \leq 2.$$

We only need to show that there exists ω_1 such that $\lambda_1 \geq \lambda$ and $\mu_1 \geq \mu$ where λ_1, μ_1 are the respective margin at ω_1 for the two players. Suppose that Q_1 is obtained by replacing machine $n-1$ and n of Q by machine $(n-1)'$. Now consider the Nash equilibrium ω of Q. Consider the allocation $\omega_0 = (x_1, \ldots, x_{n-2}, x_{n-1} + x_n, y_1, \ldots, y_{n-2}, y_{n-1} + y_n)$. Let λ_0, μ_0 be the respective margin of the players on machine $(n-1)'$ at ω_0. We now show that $\lambda_0 \geq \lambda$ and $\mu_0 \geq \mu$. We wish to show that

$$(u_{n-1} + u_n) \frac{y_{n-1} + y_n}{(x_{n-1} + x_n + y_{n-1} + y_n)^2} \geq u_{n-1} \frac{y_{n-1}}{(x_{n-1} + y_{n-1})^2} = u_n \frac{y_n}{(x_n + y_n)^2}$$

Let $a_{n-1} = x_{n-1} + y_{n-1}$, $a_n = x_n + y_n$, and $c = \frac{u_{n-1}}{u_n}$. Then the above is equivalent to show that

$$\frac{y_{n-1} + y_n}{(a_{n-1} + a_n)^2} \geq \min\left(c\frac{y_{n-1}}{a_{n-1}^2}, (1-c)\frac{y_n}{a_n^2}\right).$$

The right hand side achieves maximum at

$$\frac{y_{n-1}y_n}{y_{n-1}a_n^2 + y_n a_{n-1}^2},$$

when

$$c = \frac{y_n}{a_n^2} / (\frac{y_{n-1}}{a_{n-1}^2} + \frac{y_n}{a_n^2}).$$

It is now easy to verify that

$$\frac{y_{n-1} + y_n}{(a_{n-1} + a_n)^2} \geq \frac{y_{n-1}y_n}{y_{n-1}a_n^2 + y_n a_{n-1}^2}.$$

For all the other machines, they still have margin λ and μ. By Lemma 7, there exists a Nash equilibrium of Q_1 with margin λ_1, μ_1 with $\lambda_1 \geq \lambda$ and $\mu_1 \geq \mu$. This proves Lemma 8. □

Divide the machines into two sets $L_1 = \{i \,|\, u_i \geq w_i\}$ and $L_2 = \{i \,|\, u_i < w_i\}$. We can then merge all the machines in L_1 into one machine and all the machines in L_2 into the other. This way, the social optimum remains the same. By Lemma 8, the merging only reduces $U(Q)$ and therefore the efficiency π. This proves Lemma 4. □

B Proof of Lemma 5

Proof: Suppose that the two users' weights are, respectively, $(\alpha, 1-\alpha)$, $(\beta, 1-\beta)$ with $0 < \alpha, \beta < 1$. And their allocation is $(x, 1-x)$ and $(y, 1-y)$. Let $s = x + y$ and $\tau = (2-s)/s$. We have the following equalities.

$$s = 2/(1+\tau) \tag{2}$$

$$U = 2 - 2(\lambda + \mu) \tag{3}$$

$$\lambda + \mu = \frac{\alpha\beta}{s} + \frac{(1-\alpha)(1-\beta)}{2-s} \tag{4}$$

Equality (2) is obvious. Equality (3) is proved in the proof of Lemma 8. Equality (4) is the consequence of the following equations:

$$\frac{\lambda}{\alpha} + \frac{\mu}{\beta} = \frac{1}{s} \tag{5}$$

$$\frac{\lambda}{1-\alpha} + \frac{\mu}{1-\beta} = \frac{1}{2-s} \tag{6}$$

Equation (5) is due to $\alpha\frac{y}{s^2} = \lambda$, $\beta\frac{x}{s^2} = \mu$, and $s = x + y$. Clearly, (4) follows if we multiply (5) by $\alpha\beta$ and (6) by $(1-\alpha)(1-\beta)$ and then add them.

Now, we define some notations for later use.

$$
\begin{aligned}
b &= \frac{1}{4\alpha\beta} + \frac{1}{4\beta} + \frac{7}{4\alpha} - 2 \\
c &= \frac{1}{2\alpha} + \frac{1}{2\beta} - 1 \\
d &= \frac{(1-\alpha)(1-\beta)}{\alpha\beta} \\
e &= b - c = \frac{5}{4\alpha} + \frac{1}{4\alpha\beta} - \frac{1}{4\beta} - 1
\end{aligned}
$$

Since $0 < \alpha, \beta < 1$, it is easy to verify that $b, c, d, e > 0$.

From

$$
\begin{aligned}
\alpha\frac{y}{s^2} &= (1-\alpha)\frac{1-y}{(2-s)^2}\,, \\
\beta\frac{x}{s^2} &= (1-\beta)\frac{1-x}{(2-s)^2}\,,
\end{aligned}
$$

we have that

$$
\begin{aligned}
\alpha\tau^2 y &= (1-\alpha)(1-y)\,, \\
\beta\tau^2 x &= (1-\beta)(1-x)\,.
\end{aligned}
$$

That is, $x = \frac{1}{1+\tau^2(1-\beta)/\beta}$ and $y = \frac{1}{1+\tau^2(1-\alpha)/\alpha}$. Further, $x + y = s = \frac{2}{1+\tau}$. Therefore,

$$
\frac{1}{1+\tau^2(1-\beta)/\beta} + \frac{1}{1+\tau^2(1-\alpha)/\alpha} = \frac{2}{1+\tau}\,.
$$

Simplifying the above equation, we obtain that

$$
\tau^3 - c\tau^2 + c\tau - d = 0\,. \tag{7}
$$

Without loss of generality, we assume that $\alpha \geq \beta, 1-\beta, 1-\alpha$. Thus, $U^* = \alpha + 1 - \beta$. Combining Equalities (2,3,4), we have that

$$
\begin{aligned}
U &= 2 - 2(\lambda + \mu) \\
&= 2 - 2\left(\frac{\alpha\beta}{s} + \frac{(1-\alpha)(1-\beta)}{2-s}\right) \\
&= U^* - (\alpha\beta\tau + \frac{(1-\alpha)(1-\beta)}{\tau} + 2\alpha\beta - 2\beta)\,.
\end{aligned}
$$

Now, we assume that $U/U^* < 3/4$ and derive contradiction. If $U/U^* < 3/4$, then

$$\alpha\beta\tau + \frac{(1-\alpha)(1-\beta)}{\tau} + 2\alpha\beta - 2\beta > (1+\alpha-\beta)/4\,. \tag{8}$$

The above inequality is equivalent to that $F_{\alpha,\beta}(\tau) > 0$, where

$$F_{\alpha,\beta}(\tau) = \tau^2 - b\tau + d\,.$$

We now just need to show that for any $0 < \beta \leq \alpha < 1$, there does not exist positive root for (7) which satisfies $F_{\alpha,\beta}(\tau) > 0$. The proof is by algebraic arguments, and the details are omitted. $\square$