Capturing Resource Tradeoffs in Fair Multi-Resource Allocation

Doron Zarchy, David Hay, *Member, IEEE* and, Michael Schapira, *Member, IEEE*

***Abstract*— Cloud computing platforms provide computational resources (CPU, storage, etc.) for running users' applications. Often, the same application can be implemented in several possible ways, each with different resource requirements. Taking advantage of this flexibility when allocating resources to users can both benefit users and lead to better global resource utilization. We develop a framework for fair resource allocation that captures such implementation tradeoffs by allowing users to submit multiple "resource demands". We present a general impossibility result for resource allocation with such resource-tradeoffs: no mechanism can simultaneously satisfy Pareto optimality, strategyproofness, and envy-freeness. We then present and analyze two mechanisms for fairly allocating resources in such environments: the Lexicographically-Max-Min-Fair (LMMF) mechanism and the Nash-Bargaining (NB) mechanism. We prove that NB has many desirable properties, including Pareto optimality and envy freeness, in a broad variety of environments, whereas LMMF fares better, and is even immune to manipulations, in restricted settings of interest.**

I. Introduction

How to fairly allocate resources to multiple interested parties is an age-old challenge and a prominent research area in game theory, economics, and computer science. Of special interest, from a networking perspective, is the allocation of computational resources (e.g., CPU, memory, storage, bandwidth, etc.) in cloud computing platforms. Indeed, previous studies have investigated schemes for fairly allocating multiple resources motivated by the allocation of "bundles" of heterogeneous resources in datacenters (see, e.g., [1], [2] and references therein).

Our focus here is on an unexplored aspect of resource allocation in large-scale computational environments, e.g., cloud computing platforms. Often, the same computational task can be implemented in several different ways, each with different resource requirements. Consider, e.g., the well-studied tradeoffs in task execution between the amount of CPU and the amount of memory allotted to executing a task [3]. We argue that this flexibility can be of great importance from a fair multi-resource allocation perspective, both from the individual user's perspective and from a global resource utilization perspective.

To see this, consider even the simple toy example in which a *single* user needs to run two identical tasks on a server in the cloud and no other users are competing with it over the server's resources. To execute each of the two tasks, the user needs either a large quantity of CPU and little memory, or a large quantity of memory and little CPU. Specifically, to run a task the user needs either a $(1-\varepsilon)$-fraction of the cloud's CPU and an ε-fraction of the memory *or* an ε-fraction of the CPU a $(1-\varepsilon)$-fraction of the memory. Now, if the user is limited by the cloud tenant-provider interface to only specifying a single resource requirement (as in, e.g., [1]), and chooses to report, say the much-CPU-little-memory requirement, she cannot hope to be able to complete more than a single task. Contrast this with the scenario that the user can specify multiple resource demands corresponding to different task implementations. Now, the user can specify both possible resource-requirements and consequently complete both tasks concurrently.

This toy example illustrates how exploiting the flexibility afforded by the ability to run different realizations of the same task can lead to higher utility for the user and better global utilization of resources. These effects can be greatly amplified when there are multiple users with many diverse tasks to run. Exploiting resource-tradeoffs in task implementation to better user experience and resource utilization is yet another potential gain from rendering datacenters more predictable by extending the tenant-provider interface (see, for instance, [4]).

We formally model multi-resource environments with resource-tradeoffs. Intuitively, each user is allowed to specify multiple resource-requirements (corresponding to the requirements of different task implementations) and the utility a user derives from the resources allocated to her is the maximum number of tasks she can complete with these resources. We study mechanisms for fairly allocating resources from three main angles:

- **Computational efficiency.** Does the mechanism run in time that is polynomial in the natural parameters of this setting, such as the number of users, number of resources, etc.?
- **Fairness.** Does the mechanism *fairly* allocate resources to users? We consider several well-studied notions of fairness: Pareto optimality, envy-freeness, and max-min fairness.
- **Incentives.** Are users incentivized to report their true resource requirements to the mechanism, or can a user gain by "lying" about its resource requirements?

We first present a general impossibility result: no mechanism can simultaneously be Pareto optimal, envy-free, and incentive compatible. Given this limitation, our best hope is to design a mechanism that attains two of these properties or, alternatively, impose restrictions on the users' demands so as to circumvent this impossibility result.

D. Zarchy, D. Hay, and M. Schapira are from the School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel (e-mail: doronz@cs.huji.ac.il; dhay@cs.huji.ac.il; schapiram@cs.huji.ac.il).

We present and study two mechanisms: the Nash Bargaining (NB) mechanism and the Lexicographically Max-Min Fair (LMMF) mechanism. We analyze NB and LMMF in two environments: (1) when no restrictions whatsoever are imposed on users' resource demands; and (2) when resource-tradeoffs are linear, i.e., when the total amount of resources needed to execute a task is constant, but different combinations of resources are possible (as in the above toy example). We present both positive and negative results for many different desiderata (including computational efficiency, Pareto optimality, envy freeness, strategyproofness, and more). Our results establish that while NB provides significant benefits in general, LMMF is more appealing when resource-tradeoffs are linear. We view our contributions as the first step in the exploration of how resource-tradeoffs can be leveraged to improve computing platforms. Analyzing other mechanisms and exploring other restrictions on resource-tradeoffs are left as two important directions for future research.

II. Related Work

Fairly allocating resources to different parties is a prominent research area in game theory and economics, and has also received much attention in computer science. Much of this investigation has focused on the allocation of a single resource (e.g., the famous "cake cutting" setting [5], [6]), and on allocating multiple identical units of the same resource (e.g, in scheduling [7], [8], [9], and when allocating link bandwidth [10], [11], [12]).

While systems like Dryad [13] and Hadoop [14] involve scheduling multiple heterogenous resources, these mechanisms model the multiple resources as a single resource, and consequently sometimes inefficiently utilize resources.

Ghodsi et al. [1] provided a general framework for heterogeneous resource allocation to users with different demands. [1] presents the so called "Dominant Resource Fairness" (DRF) mechanism, which equalizes the dominant resource (i.e., the most demanded resource) across users, and proves that DRF satisfies strategyproofness, envy-freeness, Pareto optimality, and sharing incentive. Parkes et al. [15] later proved that DRF actually satisfies group strategyproofness—a stronger notion of incentive compatibility. Dolev et al. [16] proposed a different notion of fairness for heterogeneous resources called "no justified complaints" (NJC) and proved the guaranteed existence of a fair allocation in the NJC sense. Gutman and Nisan [17] generalized the notion of DRF and introduced polynomial-time algorithms for computing fair allocations for both this generalized notion and for NJC. Wang et al. [18] proposed DRFH, a generalization of DRF to an environment with multiple heterogeneous servers, and showed that DRFH satisfies Pareto optimality, envy freeness, and strategyproofness, as well as other interesting properties. Parkes et al. [19] showed that DRF generalizes to more expressive settings and studied the relation between social welfare and properties such as truthfulness; They showed that DRF performs poorly in terms of social welfare, but that this is unavoidable. Chowdhury et al. [20] explored how to optimally provide isolation guarantees in multi-resource environments where a tenant's demands for different resources (links) are elastic (unlike DRF that assumes finite demands). [20] presents an allocation mechanism that maximizes network utilization and is strategyproof. [21] provide a resource allocation policy which guarantee Resource Elasticity Fairness algorithm to find fair allocations that ensure sharing incentives, envy-freeness, Pareto efficiency, and strategy proofness in large system. [22] introduce the objective of Bottleneck Max Fairness and show that fair allocation policies with this objective has favorable efficiency-fairness tradeoff over DRF.

Lan et al. presented an axiomatic approach to measuring fairness in a single-resource domain[23]. This was generalized in [2] to heterogeneous resources by considering two measures of fairness, GFJ and FDS, where GFJ measures fairness in terms of number of jobs allocated to each user and FDS measures fairness in terms of the relative size of the dominant share. [2] presents conditions that yield Pareto optimality, sharing incentive, and envy freeness.

[24] propose a sharing policy for data centers where jobs are incentivized to share the datacenter, and benefit from reporting demands and constraints truthfully. [25] propose an allocation mechanism that extends the notion of DRF to accommodate an external resource which guarantees envy-freeness, Pareto optimality, and strategy-proofness.

Grandl et al. presented Tetris, a multi-resource cluster scheduler for task arrivals where machine availability changes in an online manner. Tetris improves average job completion time by preferentially serving jobs that have less remaining work [26]. Friedman et al. [27] investigated the allocation of cash memory to heterogeneous users. [27] demonstrated that mechanisms blocking agents from accessing parts of the memory can achieve improved efficiency guarantees, despite the inherent inefficiencies of blocking.

Procaccia et al. [28] considered the problem of fairly allocating indivisible goods, focusing on the notion of maximin share (MMS) guarantee: each player's value for her allocation should be at least as high as her value if items were shared equally between players. For some valuation functions, [28] shows that such allocations may not exist, but allocations guaranteeing each player at least two thirds of her value if items were shared equally between players always exist, and can be computed in polynomial time. Kurokawa et al. [29] designed an algorithm that provably finds an MMS allocation with high probability when valuations are drawn at random.

[30] propose a novel market-based resource allocation framework in which the services act as buyers and fog resources act as divisible goods in the market which aims to compute a market equilibrium (ME) solution at which every service obtains its favorite resource bundle under the budget constraint, while the system achieves high resource utilization. This equilibrium is ensures envy-freeness and sharing-incentive. [31] present a sharing allocation mechanism in a data center that uses Cobb-Douglas preferences to determine user's fair share of hardware. This mechanism guarantees sharing incentives, envy-freeness, and Pareto efficiency as well as strategy-proofness with a performance penalties of less than 10% throughput loss (in terms of instructions committed per cycle), relative to an unfair mechanism. [32] identify

a limitation in DRF that it may result in a poor resource utilization and propose a server-based approach to overcome it. Here each server allocates resources by maximizing a per-server utility function. This ensure envy-freeness, sharing incentive, bottleneck fairness, and Pareto optimality. They show how the proposed mechanism could be implemented in a distributed fashion.

Two notions of fairness that play the key roles in our work are (lexicographic) max-min fairness and proportional fairness. Max-min fairness is a classical notion that dates back to Rawls [33]. Algorithms that implement max-min fairness include various round robin schemes, proportional resource sharing [34], weighted fair queuing [35] and bandwidth allocation [36]. Other notions of fairness have also been considered in networking contexts, e.g., Foster et al. [37] proposed several fairness criteria for network allocation such as min guarantee (guaranteeing minimal bandwidth for every virtual machine), and payment proportionality (where bandwidth allocation is based on payments), and presented different mechanisms. Nash Bargaining was introduced by John Nash [38] as a general approach to collaboration in environments with self-interested parties that satisfies several axioms, including Pareto optimality. Nash bargaining coincides with the well-studied notion of proportional fairness and was shown to coincide with market equilibria for a large class of utility functions [39], [40]. Cole, et al. proposed a truthful mechanism that approximates the Nash Bargaining solution [41], but at the cost of "throwing away" a large fraction of the resources. Caragiannis et al. [42] studied the maximum Nash welfare (MNW) solution and show that MNW solution is fair and proved that it selects allocations that are envy free up to one good.

A preliminary version of this paper appeared in the proceedings of INFOCOM 2015 [43]. We point out that this preliminary version did not contain the general impossibility result showing no mechanism can simultaneously be Pareto optimal, envy-free, and incentive compatible. In addition, we present here a more thorough exposition of the proofs of our other main results and a more detailed treatment of the related work.

III. Model and Desiderata

A. Model

Users, resources, and resource-demands. A cloud computing environment provides a pool of k computational resources. Let C_r denote the available quantity of resource r. A set $N = \{1, ..., n\}$ of users shares the cloud's resource pool. Each user $j \in N$ has a task to perform that can be implemented in M_j different ways. The resource requirements for j's task are thus captured by a set of M_j *resource-demands* $D_j = \{d_{j1}, ..., d_{jM_j}\}$, where each element in D_j is a k-dimensional *demand vector* d_{jm} that specifies the quantity of each of the k resource required for the m'th implementation of the task. E.g., if the set of resources consists of CPU and memory only, an implementation that requires 1 unit of CPU and 3 units of memory is represented by the demand vector (1,3). Let d^r_{jm} denote the r'th resource in demand vector d_{jm}.

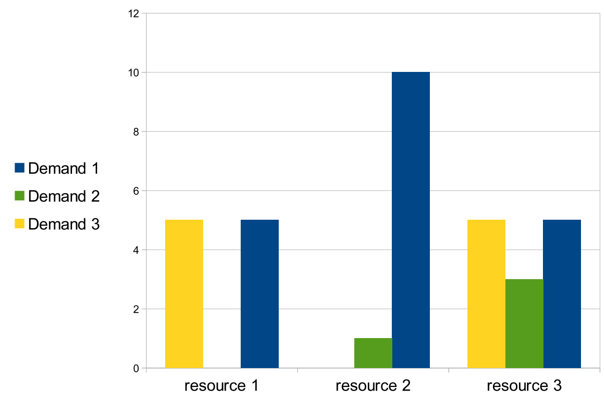

Fig. 1. Consider a resource pool of $(11, 11, 13)$ and a user with three demand vectors $d_{11} = (1, 2, 1)$, $d_{12} = (0, 1, 3)$, and $d_{13} = (2, 0, 2)$. The optimal "packing" of the user's demands in this resource pool is $5 \times (1, 2, 1) + 1 \times (0, 1, 3) + 2.5 \times (2, 0, 2) \prec (11, 11, 13)$, thus yielding a utility of 8.5, as described above.

For any two vectors $X, Y \in \mathbb{R}^n$, let $X > Y$ denote that for every component $r \in [1, n]$, $X^r \geq Y^r$, and there is at least one component for which the inequality is strict. $X < Y$ is defined analogously.

Utility functions. Each user j has a *utility function* (or utility, in short) u_j such that, for every vector of resource quantities $X = (X_1, \ldots, X_k)$, $u_j(X)$ specifies the utility user j derives from being allocated these quantities. Our focus here is on the natural "maximum packing" utility function, which captures the number of tasks the user can execute with its allocated resources. Before formally presenting this utility function, consider the example in Figure 1). User 1 has demands $D_1 = \{(1, 2, 1), (0, 1, 3), (2, 0, 2)\}$. Suppose that 1 is allocated the vector of resource quantities $X = (11, 11, 13)$. Then, user 1's utility is 8.5 as this is the maximum amount of tasks user 1 can complete with these resource quantities, computed as follows: $5 \times (1, 2, 1) + 1 \times (0, 1, 3) + 2.5 \times (2, 0, 2) \prec (11, 11, 13)$. Formally, if user j is allocated resources $X = (X_1, ..., X_k)$, $u_j(X)$ is the solution to the following linear program:

$$\begin{aligned} u_j(X) = \quad & \max \sum_{m=1}^{M_j} \alpha_m \qquad (1) \\ \text{subject to} \quad & \sum_{m=1}^{M_j} \alpha_m d_{jm} \leq X \\ & \alpha_m \geq 0 \quad \forall m \in [M_j] \end{aligned}$$

Mechanisms. The cloud allocates resources to the users by receiving as input users' resource-demands and then running some resource-allocation *mechanism* to compute the quantity of each resource allocated to each user. All computed allocations must be *feasible*, in the sense that the overall quantities of resources allocated cannot exceed the total amount of resources in the cloud's resource pool. Thus, a mechanism takes as input $D_j = \{d_{j1}, ..., d_{jM_j}\}$ from each user j and allocates, to each user j, a vector of resource quantities $X_j = (X_{j1}, \ldots, X_{jk})$ such that for every resource $r \in [k]$, $\sum_{j \in N} X_{jr} \prec C_r$.

B. Desiderata

We are interested in mechanisms that are (1) computationally-efficient, (2) fair, and (3) provide incentives for users to adhere to the mechanism. While computational efficiency simply means that the mechanism must run in time that is polynomial in the input parameters—the number of resources k, the number of users n, and the size of each user j's set of resource demands D_j—formulating fairness and incentives is more subtle.

Fairness. We present below three well-studied notions of fairness from economic theory:

- **Pareto-Optimality (PO):** A mechanism is PO if the allocation it outputs is such that in no other allocation does some user have strictly higher utility unless some other user has strictly lower utility, i.e., if the mechanism returns allocation $Y = (Y_1, \ldots, Y_n)$ then in any feasible allocation $X = (X_1, \ldots, X_n)$, if $u_i(X_i) > u_i(Y_i)$ for some user $i \in N$ then $u_j(X_j) < u_j(Y_j)$ for some other user $j \in N$.
- **Envy-Freeness (EF):** A mechanism is EF if it returns allocation $Y = (Y_1, \ldots, Y_n)$ such that no user strictly prefers another user's assigned resources to its own, i.e., for every pair of users $i, j \in N$, $u_i(Y_i) \geq u_i(Y_j)$.
- **Max-Min Fairness (MMF):** A mechanism is MMF if it maximizes the utility of the "least happy" user, i.e., it outputs the allocation $Y = (Y_1, \ldots, Y_n)$ for which the value $\min_{i \in N} u_i(Y_i)$ is maximized.

Incentives We next present three important concepts: strategyproofness, collusion proofness, and sharing-incentive.

- **Strategyproofness (SP):** A mechanism is SP if no user can benefit by misreporting his resource-demands regardless of other users' reports, i.e., for each user $i \in N$, and for every possible report of resource-demands D_j by every user $j \neq i$, if Y_i is the set of resources allocated to i when i reports his true resource demands D_i, $u_i(Y_i) \geq u_i(X)$ for every set of resources X that i can be allocated by reporting different resource-demands. We point out that some of our positive results actually apply to the stronger notion of "group-strategyproofness" (see, e.g., [15]).
- **Collusionproofness (CP):** CP is the extension of SP to *coalitions* of users. A mechanism is CP if there is no group of manipulators who benefit from colluding, i.e., for any coalition $S \subset N$, if Y, Y' are allocations such that Y is the mechanism's allocation when all users report their true demands, and Y' is an allocation that results from users in S misreporting demands, then under CP mechanism, if $u_i(Y_i') > u_i(Y_i)$ for some user i, there exists user $j \in S \backslash i$ such that $u_j(Y_j') < u_j(Y_j)$.
- **Sharing-Incentive (SI):** A mechanism is SI if each user (weakly) prefers the mechanism's allocation to getting a fraction of $\frac{1}{n}$ of each of the resources (his arguably "fair share"), i.e., for every user $j \in N$, $u_j(Y_j) \geq u_j(\frac{1}{n}, ..., \frac{1}{n})$, where Y_j specifies the resource quantities assigned to user j in the mechanism's outputted allocation.

Impossibility Result

We next present the following general impossibility result: any mechanism that is Pareto optimal and strategyproof cannot be envy free nor sharing incentive. We prove the following two theorems.

Theorem 1. *Any mechanism that is Pareto optimal and strategyproof is not envy free.*

Theorem 2. *Any mechanism that is Pareto optimal and strategyproof is not sharing incentive.*

We prove that the above theorem statements are true even in the highly restricted scenario in which two users $N = \{1, 2\}$ compete over the resource pool $R = (1, 1)$, and each user has a demand set of the form $D^w = \{(w, 0), (0, 1)\}$ for some $w > 0$.

Let $\mathbb{D} = \{D^{w_1}, D^{w_2} : w_1, w_2 \in \mathbb{R}_+\}$ be the input of the mechanism. For $D \in \mathbb{D}$, let $M_i(D)$ denote the resulting allocation of mechanism M to user i. We first show that the output space of any Pareto optimal (PO) mechanism is such that at least one of the users benefits by misreporting her demands, with the exception of the trivial mechanism that always allocates all the resources to the same (predetermined) user.

Definition 1. Let $Dom_1 = A_1 \cup A_2$ be set of resource vectors where $A_1 = \{(x, 0) : x \in [0, 1]\}$ and $A_2 = \{(1, x) : x \in [0, 1]\}$ Similarly, let $Dom_2 = B_1 \cup B_2$, where $B_1 = \{(0, x) : x \in [0, 1]\}$ and $B_2 = \{(x, 1) : x \in [0, 1]\}$.

The following lemma establishes that any PO mechanism outputs an allocation in Dom_1 or in Dom_2.

Lemma 1. *For any Pareto-optimal mechanism M, $M_1(D) \in Dom_1$, $M_2(D) \in Dom_2$ if $w_1 < w_2$ and $M_1(D) \in Dom_2$, $M_2(D) \in Dom_1$ if $w_1 > w_2$.*

Proof. Let M be a Pareto-optimal mechanism and D^{w_1}, D^{w_2} be the input demands of user 1 and user 2 respectively of the mechanism. Consider the case that $w_1 < w_2$. Suppose for the point of contradiction that $M_1(D) \notin Dom_1$. That implies, $M_1(D) = (x_1, x_2)$ such that $x_1 < 1$ and $x_2 > 0$ and $M_2(D) = (1 - x_1, 1 - x_2)$. Therefore the corresponding utilities are,

$$u_1 = \frac{x_1}{w_1} + x_2 \text{ and } u_2 = \frac{1 - x_1}{w_2} + (1 - x_2) \tag{2}$$

Let p be a constant such that $\frac{1}{w_2} < p < \frac{1}{w_1}$ and $t > 0$ be constant such that $t \leq (1 - x_1)$ and $tp \leq x_2$. Consider a scenario of a trade where user 1 transfers tp units of resource 2 to user 2, and in return, user 2 transfers t units of resource 1 to user 1. Then, the utility of user 1 becomes

$$u_1' = \frac{x_1 + t}{w_1} + x_2 - tp$$

and utility of user 2 becomes

$$u_2' = \frac{1 - x_1 - t}{w_2} + (1 - x_2) + tp$$

Therefore,

$$u_1' - u_1 = t(\frac{1}{w_1} - p) > 0$$

and

$$u'_2 - u_2 = t(p - \frac{1}{w_2}) > 0$$

The fact that $u'_1 > u_1$ and $u'_2 > u_2$ contradicts Pareto optimality. Therefore $M_1(D) \in Dom_1$. Since for any allocation $M_1(D) = (x_1, x_2) \implies M_2(D) = (1 - x_1, 1 - x_2)$, by using the same argument we have $M_2(D) \in Dom_2$. We now consider the case that $w_1 > w_2$. Suppose for the point of contradiction that $M_1(D) \notin Dom_2$. That implies, $M_1(D) = (x_1, x_2)$ such that $x_1 > 0$ and $x_2 < 1$. Note that $M_2(D) = (1 - x_1, 1 - x_2)$. The utilities of the user are as in (2). Let p be a constant such that $\frac{1}{w_1} < p < \frac{1}{w_2}$ and $t > 0$ be a constant such that $t \leq (1 - x_1)$ and $tp \leq x_2$. Consider a scenario of a trade where user 1 transfers t units of resource 1 to user 2, and in return, user 2 transfers tp units of resource 1 to user 1. Then, the utility of user 1 becomes

$$u'_1 = \frac{x_1 - t}{w_1} + x_2 + tp$$

and utility of user 2 becomes

$$u'_2 = \frac{1 - x_1 + t}{w_2} + (1 - x_2) - tp$$

Therefore,

$$u'_1 - u_1 = t(p - \frac{1}{w_1}) > 0$$

and

$$u'_2 - u_2 = t(\frac{1}{w_2} - p) > 0$$

The fact that $u'_1 > u_1$ and $u'_2 > u_2$ contradicts Pareto optimality. Therefore $M_1(D) \in Dom_2$. Since for any allocation $M_1(D) = (x_1, x_2) \implies M_2(D) = (1 - x_1, 1 - x_2)$, by using the same argument we have $M_2(D) \in Dom_1$. □

According to Lemma 1 the allocations in Dom_1 induce a total ordering as $(x, 0) > (y, 0)$ and $(1, x) > (1, y)$ if and only if $x > y$, and $(1, x) > (y, 0)$ for all $x, y \in (0, 1)$. Similarly, Dom_2 induces a total ordering. Therefore we have the following corollary,

Corollary 1. *Given demands of type $\mathbb{D}$, any PO mechanism induces a total ordering over the utilities the users gain, that is for any $D, D' \in \mathbb{D}$, for each user $i \in N$, if $M(D) \neq M(D')$ then $M_i(D) > M_i(D')$ or $M_i(D) < M_i(D')$.*

We next prove the main lemma needed to establish the impossibility result(s).

Lemma 2. *Any mechanism that is strategyproof and Pareto-optimal allocates all the resources to the same user.*

Proof. Let M be a PO mechanism. We now show that unless M allocates all resources to one user, M violates SP by the following cases:

1) Consider a mechanism M such that $|\{M(D^{w_1}, D^{w_2}) : w_1 \neq w_2\}| > 1$, i.e., suppose that there are at least two outputs by M for two different input demands. Let $D_a = (D^{w_1}, D^{w_2}))$, $D_b = (D^{w'^1}, D^{w'^2})$, and $D_c = (D^{w^1}, D^{w'^2})$ be three different demands. Let $A_a = M_1(D_a)$, $A_b = M_1(D_b)$, and $A_c = M_1(D_c)$ be the corresponding allocations to user 1, and suppose w.l.o.g that $A_a \neq A_b$ (which can be assumed by the case condition). By Corollary 1 we have $A_a < A_b$ or $A_a > A_b$. We claim that there is always a user that is better off by misreporting. Consider the case that $A_c = A_a$. We analyze the following cases:
   - $A_a < A_b$. Given that the true demand is D_c, then if user 1 misreports $D^{w'^1}$, then the input of the mechanism becomes D_b and the user obtains A_b (instead of A_c as under true report) and since $A_c < A_b$ (as $A_c = A_a$) the user benefits by misreporting.
   - $A_a > A_b$. Given that the true demand is D_b, then if user 1 misreports D^{w^1}, then the input to the mechanism becomes D_c and the user obtains A_c (instead of A_b under true report) and since $A_b < A_c$ (as $A_c = A_a$) the user benefits by misreporting.

   If $A_c \neq A_a$ then we can use similar arguments to show that there is always a user that is strictly better off by misreporting. We are now left with the following very restrictive mechanism:
2) Suppose that $|\{M(D^{w_1}, D^{w_2}) : w_1 < w_2\}| = 1$ and $|\{M(D^{w_1}, D^{w_2}) : w_1 < w_2\}| = 1$, i.e., the mechanism outputs the same allocation $((x^1, x^2), (y^1, y^2))$ for all input demands $D = (D^{w_1}, D^{w_2})$ such that $w_1 < w_2$, and the same allocation $(x'^1, x'^2), (y'^1, y'^2)$ for all input demands $D = (D^{w_1}, D^{w_2})$ such that $w_1 > w_2$ (that is the absolute values of w_1 and w_2 has no impact the allocation in both cases). There are three possible cases:
   a) If $x^1 > x'^1$, then there is a sufficiently **small** t such that for every $w_1 < t$, $\frac{x^1}{w_1} + x^2 > \frac{x'^1}{w_1} + x'^2$. Hence, given that under true demand $w_1 > w_2$, if $w_1 < t$ then user 1 benefits by misreporting $\tilde{w}_1 < w_2$.
   b) If $x^2 > x'^2$, then there is a sufficiently **large** t such that for every $w_1 > t$, $\frac{x^1}{w_1} + x^2 > \frac{x'^1}{w_1} + x'^2$. Similarly to the previous case, given that under true demand $w_1 > w_2$, if $w_1 > t$ then user 1 benefits by misreporting $\tilde{w}_1 < w_2$.
   c) Otherwise (that is, if $x^1 < x'^1$ and $x^2 < x'^2$), we have $y^1 > y'^1$ and $y^2 > y'^2$ (as $y^r = (1 - x^r)$ and $y'^r = (1 - x'^r)$ for $r \in 1, 2$). Therefore, given that $w_1 > w_2$, user 2 always benefits by misreporting $\tilde{w}_2 > w_1$.
3) Suppose that $|\{M(D^{w_1}, D^{w_2}) : w_1 \neq w_2\}| = 1$, i.e., there is allocation A such that for all $D \in \mathbb{D}$, $M_1(D) = A$. Suppose, w.l.o.g that $A \in Dom_1$. However, if $w_1 > w_2$ then by Lemma 1 $M_1(D) \in Dom_2$ (as M is PO). Hence, $A \in Dom_1 \cap Dom_2$. Since $Dom_1 \cap Dom_2 = \{((1, 1), (0, 0)), ((0, 0), (1, 1))\}$, it follows that the only possible mechanism that is both PO and SP is the mechanism that allocates all the resources to the same user.

□

Since, by Lemma 2, user 2 does not obtain any resource (see above), user 2 envies user 1 and Theorem 1 follows.

Since, by Lemma 2) user 2 has no incentive to use the mechanism, it is not SI. Theorem 2 follows.

IV. Two Mechanisms

We now describe two mechanisms for fairly allocating resources: the Lexicographically Max-Min Fair (LMMF) mechanism and the Nash Bargaining (NB) mechanism. These two mechanisms reflect two different economic approaches to resource allocation and have previously been studied in other contexts (e.g., when each user has a single demand vector). See Section II for an exposition of related work.

A. The LMMF mechanism

We first present the Max-Min fair (MMF) mechanism, which allocates resources so as to maximize the utility of the "poorest" user (i.e., the user who can complete the lowest number of tasks). To illustrate this, consider the following examples:

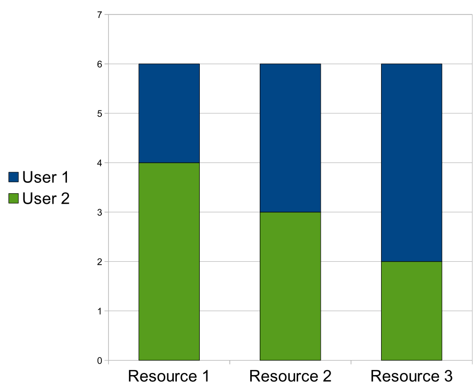

Fig. 2. Max-min fair allocation in Example 1. User 1 gets utility of 3 as it can complete 3 tasks as follows: $1\times(2,3,0)+2\times(0,0,2)$. User 2 gets utility of 3 as well as it can also complete 3 tasks as follows: $2\times(2,0,1)+1\times(0,3,0)$.

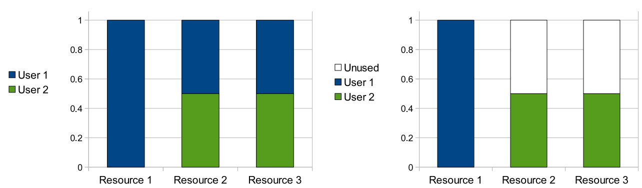

Fig. 3. Example 2. On the right-hand side is a MMF allocation. On the left-hand side is a LMMF allocation.

Example 1. *A resource pool of* $C = (6,6,6)$ *and two users with demands* $D_1 = \{(2,3,0),(0,0,2)\}$ *and* $D_2 = \{(2,0,1),(0,3,0)\}$.

Example 2. *A resource pool of* $C = (1,1,1)$ *and two user with single demand vector each,* $D_1 = \{(1,0,0)\}$ *and* $D_2 = \{(0,\frac{1}{2},\frac{1}{2})\}$.

In the scenario described in Example 1, MMF returns the allocation $Y_1 = (2,3,4)$ and $Y_2 = (4,3,2)$. Observe that the utility of both users is then 3 (user 1 can "pack" $1\times(2,3,0)+2\times(0,0,2)$ in his allocated bundle of resources, whereas user 2 can pack $2\times(2,0,1)+1\times(0,3,0)$). See Figure 2.

However, MMF can sometimes be suboptimal, in the sense that available resources that may benefit users might not be allocated. For instance, in example 2, a possible MMF allocation is $Y_1 = (1,0,0), Y_2 = (0,\frac{1}{2},\frac{1}{2})$. Observe that this allocation is indeed MMF as under any feasible allocation user 1 cannot hope to achieve utility greater than 1. Hence, any allocation under which user 1's utility is 1 and user 2's utility is at least 1 is MMF. However, this allocation is not Pareto optimal, as doubling user 2's resources will increase his utility without harming user 1. This is where *lexicographic* max-min fairness (LMMF) comes in.

To overcome the suboptimality of MMF, LMMF also maximizes the utility of the "poorest" user but, amongst all such allocations, selects the one that maximizes the utility of the second poorest user, and so on. In Example 1, LMMF returns the exact same allocation as MMF. However, in Example 2, LMMF outputs the allocation $Y_1 = (1,0,0)$ and $Y_2 = (0,1,1)$, which is indeed Pareto optimal. See Figure 3.

We are now ready to formally define MMF and LMMF.

The Max-Min Fair (MMF) mechanism. Given a resource pool $C = (C_1,\ldots,C_k)$ and users' resource demands, MMF finds an allocation $Y = (Y_1,...,Y_n)$,

$$\begin{aligned} \text{Maximize} \quad & t && (3)\\ \text{Subject to} \quad & u_j(Y_j) \geq t \quad \forall j \in N \\ & \sum_{j\in N} Y_j \prec C && (4) \end{aligned}$$

The Lexicographically Max-Min Fair (LMMF) mechanism. To formally present the LMMF mechanism we require the following terminology and notation. For a given allocation $Y = (Y_1,\ldots,Y_n)$, let $\langle Y\rangle$ denote the vector that contains the n elements in $\{u_1(Y_1),\ldots,u_n(Y_n)\}$ sorted in non-decreasing order, i.e., $u_{i_1}(Y_{i_1}) \leq u_{i_2}(Y_{i_2}) \leq \ldots u_{i_n}(Y_{i_n})$. An allocation Y is *lexicographically greater* than another allocation Y', denoted by $\langle Y\rangle \succ \langle Y'\rangle$, if the first non zero component of $(\langle Y\rangle - \langle Y'\rangle)$ is positive. An allocation vector Y is *lexicographically no less than* Y', denoted by $\langle Y\rangle \succeq \langle Y'\rangle$, if $(\langle Y\rangle - \langle Y'\rangle) = 0$, or the leftmost non-zero component of $(\langle Y\rangle - \langle Y'\rangle)$ is positive. We are now ready to define lexicographic max-min fairness.

Definition 2. An allocation Y is lexicographically max-min fair if $\langle Y\rangle \succeq \langle Y'\rangle$ for every feasible allocation Y'.

The LMMF mechanism outputs, for every input sets of resource-demands, a lexicographically max-min allocation. We now explain how this computation is executed. We prove that the computation indeed terminates in polynomial time and outputs a lexicographically max-min fair allocation in Section V.

LMMF proceeds in iterations:

Iteration 1: LMMF solves a linear program to compute the maximum value a_1 such that in some feasible allocation $Y =$

$(Y_1, \ldots, Y_n)$ the utility of each and every user is *exactly* a_1.[1] LMMF then checks, for every user, whether his utility in Y cannot be increased without decreasing the utility of other users. All such users are placed in the set p_1.

Iteration 2: LMMF solves a linear program to compute the maximum value a_2 such that in some feasible allocation $Y = (Y_1, \ldots, Y_n)$ the utility of each user in p_1 is exactly a_1 and the utility of all other users is exactly a_2. LMMF then checks, for every user not in p_1, whether his utility in Y cannot be increased without decreasing the utility of users in p_1. All such users are placed in the set p_2.

Iteration t=3,4,. . . : Similarly, LMMF solves a linear program to compute the maximum value a_t such that in some feasible allocation $Y = (Y_1, \ldots, Y_n)$ the utility of each user in p_i for all $i < t$ is exactly a_i and the utility of all other users is exactly a_t. LMMF then checks, for every user not in p_i for $i < t$, whether his utility in Y cannot be increased without decreasing the utility of users in p_i for $i < t$. All such users are placed in the set p_t.

This continues until all users are placed in some p_t set, at which point LMMF outputs the allocation Y computed at the last iteration.

To see how LMMF works in a concrete scenario, consider resource pool $C = (1, 1, 1, 1)$ and three users with demands $D_1 = \{(\frac{1}{2}, \frac{1}{2}, 0, 0)\}$, $D_2 = \{(0, \frac{1}{2}, \frac{1}{2}, 0)\}$, and $D_3 = \{(\frac{1}{2}, 0, \frac{1}{2}, 0), (0, 0, 0, 1)\}$. At the first iteration, LMMF solves a linear program to compute $a_1 = 1$, as all users can achieve utility of 1, e.g., in the feasible allocation $Y = ((\frac{1}{2}, \frac{1}{2}, 0, 0), (0, \frac{1}{2}, \frac{1}{2}, 0), (0, 0, 0, 1))$ where users 1 and 2 are each allocated precisely their demand vectors and user 3 is allocated his second demand vector. As users 1 and 2 cannot attain utility higher than 1 in Y (as the second resource is fully utilized) LMMF creates the set $p_1 = \{1, 2\}$. In the next iteration, LMMF solves another linear program to compute $a_2 = 2$. Indeed, consider the allocation resulting from allocating users 1 and 2 the exact same resources as in Y and adding $(\frac{1}{2}, 0, \frac{1}{2}, 0)$ to 3's allocated resources in Y. Observe that again users 1 and 2 have a utility of 1, but now user 3's utility is 2 (=a_2). User 3 is now placed in the set p_2. As all users are now placed in either p_1 or p_2, LMMF terminates and outputs this allocation.

B. Nash Bargaining Mechanism

Consider a scenario in which each user initially has an "endowment" of a $\frac{1}{n}$'th fraction of each resource. Suppose that there are only two resources, each of quantity 1, and two users, 1 and 2, with demands $D_1 = \{(1, 0)\}$ and $D_2 = \{(0, 1)\}$. Clearly, the users' initial endowments of $(\frac{1}{2}, \frac{1}{2})$ are not Pareto optimal, in the sense that both users are better off if the entire first resource is allocated to user 1 and the entire second resource is allocated to user 2. Much research in game theory and economics studies how different strategic agents should cooperate when non-cooperation leads to Pareto suboptimal results. The Nash Bargaining (NB) mechanism implements one solution to this problem, given by John Nash [38].

Intuitively, NB allocates resources so as to maximize the product of all users' utilities. Consider, for instance, a resource pool of $(8, 8)$ and two users with demands $D_1 = \{(5, 1), (2, 2)\}$ and $D_2 = \{(1, 4), (2, 2)\}$. The allocation that maximizes the product of utilities is allocating $(4, 4)$ to each user, as in this scenario each user has a utility of 2 (as two tasks can be executed by each user) and the product gives $2 \cdot 2 = 4$ (and it can be verified that no other allocation leads to a higher value). Formally, NB outputs the allocation $Y = (Y_1, ..., Y_n)$ such that

$$\max \prod_{j \in N} u_j(Y_j) \tag{5}$$

$$\sum_{j \in N} Y_j \leq C \tag{6}$$

where $C = (C_1, \ldots, C_n)$.

V. Results for Unrestricted Demands

We now explore the guarantees of the two mechanisms presented in Section IV for general resource demands, i.e., when no restrictions whatsoever are imposed on users' resource demands. We consider the three main criteria presented in Section III: computational efficiency, fairness, and incentive compatibility. We then discuss the other desiderata presented in Section III: non-wastefulness and sharing incentive. The following table summarizes our results for general demands.

	CE	EF	SI	PO	LMMF	SP
LMMF	✓			✓	✓	
NB	✓	✓	✓	✓		

(CE = computationally efficient, EF = envy free, SI = sharing incentive, PO = Pareto optimal, LMMF = lexicographically max-min fair, SP = strategyproof)

A. Computational Efficiency

Proposition 1. *LMMF allocation can be computed efficiently*

Proof. The LMMF mechanism (see subsection IV-A), solves at most n linear programs (as the mechanism creates a set p_t at each iteration t, such that $p_1, ..., P_T$ partitions the set of users N), where each can be computed in time that is polynomial in $|N|$, k, and $\max_{j \in N} M_j$. □

Proposition 2. *NB allocation can be computed efficiently*

Proof. An explicit way of writing (5) is,

$$\max \prod_{j \in N} \left(\max \sum_{m=1}^{M_j} \alpha_{jm} \right) \tag{7}$$

$$\sum_{j \in N} \sum_{m=1}^{M_j} \alpha_{jm} d_{jm} \leq C \tag{8}$$

$$\alpha_{jm} \geq 0 \quad \forall m \in [M_j], \forall j \in N$$

Fixing some allocation Y_j to user j, the coefficients $\alpha_{j1}, ..., \alpha_{jM}$ do not influence the coefficients $\alpha_{i1}, ..., \alpha_{iM}$ of

[1]We point out that this can indeed be formulated as a linear program (for the interest of brevity, the formulation is deferred to the full paper.)

any other allocation $Y_i \neq Y_j$. Therefore, we can simplify the objective function of the NB program to,

$$\max \prod_{j \in N} \sum_{m=1}^{M_j} \alpha_{jm} \tag{9}$$

Transforming the objective function of program 9 into $\max \sum_{j \in N} \log \sum_{m=1}^{M_j} \alpha_{jm}$ (while not changing the constraints) yields the same solution (as such a transformation does not affect the solution of a convex program). Since this is a convex function, the solution can be efficiently computed. □

B. Fairness

LMMF. We first analyze the fairness properties of LMMF. The first step is proving that LMMF indeed computes a lexicographically max-min fair allocation. We then observe that LMMF is Pareto optimal and prove that it is not, however, envy free.

The nontrivial proof that the LMMF mechanism, as defined in Section IV, indeed computes a lexicographically max-min fair allocation, follows from a combination of lemmas.

Claim 1. *For any two feasible allocations Y, Y' and $\gamma \in [0,1]$, the allocation $\hat{Y} = (\gamma Y + (1-\gamma)Y')$ is feasible.*

Proof. For every resource $r \in R$ we have,

$$\begin{aligned}\sum_{j=1}^{n} \hat{Y}_j &= \sum_{j=1}^{n} (\gamma Y_j + (1-\gamma)Y'_j) \\ &= \gamma \sum_{j=1}^{n} Y_j + (1-\gamma)\sum_{j=1}^{n} Y'_j \leq \gamma C + (1-\gamma)C = C\end{aligned}$$

where the inequality follows from the fact that Y and Y' are feasible allocations. □

Lemma 3. *(Continuity) For every $j \in N$ and two resource vectors X and X', the function $f_j(\gamma) = u_j(\gamma X + (1-\gamma)X')$ is continuous on $\gamma \in [0,1]$.*

Proof. Let $\Delta = X - X'$ and $\gamma \in (0,1)$. We need to show that for every $\epsilon > 0$ however small, there exists $\delta > 0$ such that $|f_j(\gamma+\delta) - f_j(\gamma)| < \epsilon$. Note that $|((\gamma+\delta)X + (1-(\gamma+\delta))X') - (\gamma X + (1-\gamma)X')| \leq |\delta \cdot \Delta|$. Let e_r be the unit vector $(0,...,0,1,0...,0)$, where the value of 1 is at the r'th index. By changing that allocation from X to X' the quantity of the r'th resource increases by at most $\delta \cdot \Delta^r$, where Δ^r denotes the r'th component in Δ. Let $l = \min_{m,r} d^r_{j,m}|_{l>0}$ (i.e., l is the minimal positive component across all demand vectors in D_j). Then, for any $r \in [1,k]$, if we add $e_r \cdot \delta \cdot \Delta^r$ to the allocation $\gamma X + (1-\gamma)X'$, the utility would increases by at most $\delta \frac{\Delta^r}{l}$. Therefore, with respect to all components $r \in R$ we have that

$$\begin{aligned}|f_j(\gamma+\delta) - f_j(\gamma)| &= |u_j(\gamma X + (1-\gamma)X' + \delta\Delta) - f_j(\gamma)| \\ &\leq \left|\frac{\delta |R| \Delta^*}{l}\right|\end{aligned}$$

where $\Delta^* = \max_{r \in R} \Delta^r$. Thus, setting $\delta = \frac{\epsilon \cdot l}{|R|\Delta^*}$ concludes the proof. □

Lemma 4. *The utility function $u_j(x)$ is concave in x for any $j \in N$, that is for two resource vectors X and X', and $\gamma \in [0,1]$, $u_j(\gamma X + (1-\gamma)X') \geq \gamma u_j(X) + (1-\gamma)u_j(X')$.*

The intuition behind this is that if j obtains the allocation $\gamma X + (1-\gamma)X'$ then it can always gain the utility of $\gamma u_j(X) + (1-\gamma)u_j(X')$ by using the support demand vectors (the demand vectors that are used for computing the utility) of $u_j(X)$ and $u_j(X')$ in the right proportions.

Proof. Let $\alpha_1, ..., \alpha_M$ be the coefficients of the demand vectors correspond to $u_j(X)$ (that is, $u_j(X) = \sum_{m=1}^{M} \alpha_m$). Similarly, let $\beta_1, ..., \beta_M$ and $c_1, ..., c_M$ be the coefficients of the demand vectors correspond to $u_j(X')$ and $u_j(\gamma X + (1-\gamma)X')$ respectively, for some $\gamma \in [0,1]$. By allocation constraints, $\sum_{m=1}^{M} \alpha_m d^r_{j_m} \leq X$ and $\sum_{m=1}^{M} \beta_m d^r_{j_m} \leq X'$. Therefore, $\gamma \sum_{m=1}^{M} \alpha_m d_{j_m} + (1-\gamma)\sum_{m=1}^{M} \beta_m d_{j_m} \leq \gamma X + (1-\gamma)X'$. Hence,

$$\sum_{m=1}^{M} (\gamma \alpha_m + (1-\gamma)\beta_m) d_{j_m} \leq \gamma X + (1-\gamma)X' \tag{10}$$

On the other hand, according to the constraints of $u_j(\gamma X + (1-\gamma)X')$,

$$\sum_{m=1}^{M} c_m d_{j_m} \leq \gamma X + (1-\gamma)X' \tag{11}$$

Since by definition of utility function (19), $c_1, ..., c_M$ maximize the packing of the demand vectors in D_j under constraint (11), and as the right hand side in (11) is equal to that in (10) we have,

$$\sum_{m=1}^{M} c_m \geq \sum_{m=1}^{M} \gamma \alpha_m + (1-\gamma)\beta_m \tag{12}$$

Therefore,

$$\begin{aligned}&u_j(\gamma X + (1-\gamma)X') &(13)\\ &\triangleq \sum_{m=1}^{M} c_m &(14)\\ &\geq \sum_{m=1}^{M} \gamma \alpha_m + (1-\gamma)\beta_m &(15)\\ &= \gamma \sum_{m=1}^{M} \alpha_m + (1-\gamma)\sum_{m=1}^{M} \beta_m &(16)\\ &\triangleq \gamma u_j(X) + (1-\gamma)u_j(X') &(17)\end{aligned}$$

□

We now introduce the notion of a "*pivot* user".

Definition 3. Given two allocation Y and Y', a pivot user of allocation Y with respect to Y' is the user with the lowest utility in Y that has different utility in Y'.

To illustrate that consider the example illustrated in figure 4. There are two allocations Y and Y' of four users where $\langle Y \rangle = \langle 2,3,4,5 \rangle$ where coordinates 1-4 correspond to the utilities of users i_1-i_4, respectively and $\langle Y' \rangle = \langle 2,3,4,6 \rangle$ where coordinate 1-4 correspond to users i_1, i_2, i_4, i_3, respectively.

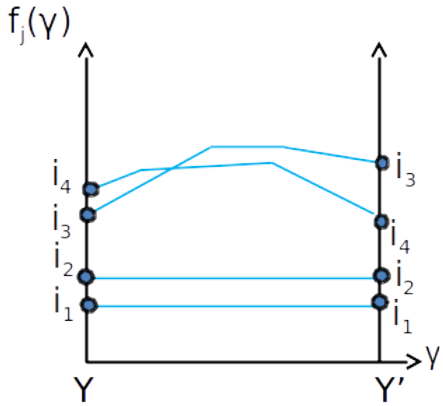

Fig. 4.
The utilities under the convex combination of allocations Y, Y'. Lemmas 3, 4 state that the convex combination of user j, $f_j(\gamma)$, is continuous and concave.

Observe that in this scenario the pivot of Y with respect to Y', is user i_3 (i.e., $\theta(Y', Y) = \{i_3\}$) , whereas the pivot of Y' with respect to Y is user i_4 (i.e., $\theta(Y', Y) = \{i_4\}$).

Claim 2. *Let $i \in \theta(Y, Y')$ and $j \in \theta(Y', Y)$ be two pivot users. Then, $\langle Y' \rangle \succ \langle Y \rangle$ if and only if $u_j(Y') > u_j(Y)$.*

The following auxiliary lemma compares users that are poorer than the pivot in Y to users in $\alpha Y + (1-\alpha)Y'$.

Claim 3. *Let Y, Y' be two allocations, then, for any convex combination of these allocations (i.e., $\hat{Y} = \alpha Y + (1-\alpha)Y'$), any user that is poorer than the pivot under Y obtains a utility under Y that is no higher than her utility under $\hat{Y}$.*

Proof. Let $q \in \theta(Y, Y')$ be the pivot, and $p \in N$ be a user such that $u_p(Y) < u_q(Y)$, i.e., p is poorer than q under allocation Y (e.g., see users i_1 and i_2 in figure 4). Since p is poorer than q under Y, $u_p(Y_p) = u_p(Y'_p)$. By Lemma 4, for all $\alpha \in (0, 1)$, $u_p(\alpha Y_p + (1-\alpha)Y'_p) \geq u_p(Y)$. □

The following lemma is the basis for proving Proposition 3, and strategyproofness and in section VI.

Lemma 5. *For any two allocations Y, Y' and pivot user $q \in \theta(Y, Y')$, if $u_q(Y') > u_q(Y)$, then there exists small $\epsilon > 0$ such that for allocation $\hat{Y} = (1-\epsilon)Y + \epsilon Y'$, we have $q \in \theta(Y, \hat{Y})$ and $u_q(\hat{Y}) > u_q(Y)$.*

To prove this we use the concavity of the utility function and the continuity of the function $f_j(\gamma)$ in Lemma 3. (for intuition see Figures 4), 5).

Formally,

Proof. Let $q \in \theta(Y, Y')$ be a pivot user. We define the following sets $P_i(Y) = \{j \in N | u_j(Y) < u_i(Y)\}$ (i.e., P_i is the set of all users that are poorer than i under allocation Y) and $H = \{j \in N | u_j(Y) \geq u_q(Y)\}$. Note that by the definition of H, $q \in \arg\min_{h \in H} u_h(Y)$. Let $\epsilon > 0$ be a sufficiently small scalar such that, w.l.o.g, q is still minimal among the users in H under the allocation $\epsilon Y' + (1-\epsilon)Y$, i.e., $q \in \arg\min_{h \in H} u_h(\epsilon Y' + (1-\epsilon)Y)$ (if there is more than one pivot user then we assume, w.l.o.g., that q is chosen to be the poorest user among the pivots in $\theta(Y, Y')$ under the allocation $\epsilon Y' + (1-\epsilon)Y$). We claim that there always exists such a small ϵ. This follows from Lemma 3, which states that for every user $h \in H$, each function $f_j(\alpha) \triangleq u_j(\alpha Y + (1-\alpha)Y')$ is continuous in $0 \leq \alpha \leq 1$. Let $\hat{Y} = (\epsilon Y' + (1-\epsilon)Y)$, and let $q_1 = \theta(Y, \hat{Y})$, $q_2 = \theta(\hat{Y}, Y)$. We inspect two cases, showing for each that $\langle \hat{Y} \rangle \succ \langle Y \rangle$.

1) $u_{q_1}(Y) < u_q(Y)$. Suppose, for point of contradiction, that $u_{q_1}(Y) > u_{q_2}(\hat{Y})$. Hence, $u_q(Y) > u_{q_2}(\hat{Y})$. As $u_q(Y') > u_q(Y)$, by Lemma 4, $u_q(\hat{Y}) > u_q(Y)$. Moreover, from the definition of ϵ, for every $h \in H$, $u_h(\hat{Y}) \geq u_q(\hat{Y})$, and so $u_h(\hat{Y}) > u_q(Y)$. Hence, we have that $u_{q_2}(\hat{Y}) < u_q(\hat{Y})$ and then it must be that $q_2 \notin H$. Thus, $u_{q_2}(Y) < u_q(Y)$. As $q_2 \in P_q$, it follows from Claim 3, that $u_{q_2}(\hat{Y}) > u_{q_2}(Y)$ (the strictness of the inequality follows from the fact that q_2 is a pivot user). Because of the assumption that $u_{q_1}(Y) > u_{q_2}(\hat{Y})$, we get that $u_{q_2}(Y) < u_{q_1}(Y)$. However, since $u_{q_2}(Y) \neq u_{q_2}(\hat{Y})$ and $q_2 \in P_{q_1}$ we have that $q_1 \notin \theta(Y, \hat{Y})$ — a contradiction. Therefore, $u_{q_2}(\hat{Y}) > u_{q_1}(Y)$ and by Claim 2, $\langle \hat{Y} \rangle \succ \langle Y \rangle$.
2) $u_{q_1}(Y) \geq u_q(Y)$. This implies that
$$u_p(Y) = u_p(\hat{Y}) \text{ for all } p \in P_q \tag{18}$$
Since $u_q(Y') > u_q(Y)$, by Lemma 4, $u_q(\hat{Y}) > u_q(Y)$. This implies that $q \in \theta(Y, \hat{Y})$ (q is a pivot of $(Y, \hat{Y})$ in addition to q_1). By definition of ϵ, and (18), we deduce that $q \in \theta(\hat{Y}, Y)$. As $u_q(\hat{Y}) > u_q(Y)$, by Claim 2, $\langle \hat{Y} \rangle \succ \langle Y \rangle$.

□

The following lemma provides a sufficient condition for an allocation Y to not be LMMF (see fig.5)

Corollary 2. *For any two allocations Y, Y' and pivot user $i \in \theta(Y, Y')$, if $u_i(Y') > u_i(Y)$, then Y is not LMMF.*

Proof. Let $i \in \theta(Y, Y')$ be a pivot user. By Lemma 5, there is some small $\epsilon > 0$ such that under allocation $\hat{Y} = (1-\epsilon)Y + \epsilon Y'$, $i \in \theta(Y, \hat{Y})$ and $u_i(\hat{Y}) > u_i(Y)$. Hence, by Claim 2, Y is not LMMF. □

We are now finally ready to prove the following statement.

Proposition 3. *The LMMF mechanism is lexicographically max-min fair.*

Proof. Let Y be the allocation that the mechanism returns for a given input and let Y^* be a LMMF allocation. We show this by induction on the iteration number t. Note, that if by the end iteration t (see Section IV) the mechanism created sets $p_1, ..., p_t$ (as described in Section IV), and all users in these sets have the same utility in both Y and Y^*, then at iteration $t+1$ all users in $N \setminus P$, where $P = \bigcup_{i \in [t]} p_i$ can achieve the same minimal utility under Y and Y^* as the linear program that LMMF solves at the (t+1)'th iteration. Hence, in particular, the pivot in both allocations has the same utility. Suppose, for point of contradiction, that Y is not LMMF, and let t be the iteration at which the pivot $q = \theta(Y, Y^*)$

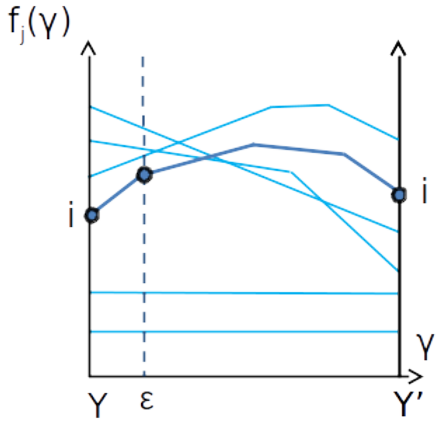

Fig. 5. The right figure illustrates the statement of Corollary 2: a sufficient condition for allocation to not be LMMF is for a pivot user $i \in \theta(Y, Y')$ to have a higher utility in Y' than in Y. This is because there exists a feasible allocation $\hat{Y} = \epsilon Y' + (1 - \epsilon)Y$, for a sufficiently small $\epsilon > 0$, in which i is a pivot user (i.e., $i \in \theta(\hat{Y}, Y)$) such that $f_i(\epsilon) > u_i(Y)$ (Lemma 5). Hence, by Claim 2, $\hat{Y} \succ Y$. Thus, Y is not LMMF.

is added to P. Then, by definition of the pivot q, it must be that $u_q(Y^*) > u_q(Y)$. Thus, by Lemma 5, there is an allocation $\hat{Y} = \epsilon Y^* + (1 - \epsilon)Y$ such that $u_q(Y) < u_q(\hat{Y})$. However, this contradicts the fact that the linear program of the LMMF mechanism maximizes the utility of all users in $N \backslash P$ at iteration t. □

We next show that LMMF is Pareto optimal.

Claim 4. *LMMF is Pareto optimal.*

Proof. Let Y be an output allocation of some LMMF mechanism. Suppose that Y is not Pareto optimal and so there is allocation Y' such that for all $i \in [1, n]$, $u_i(Y') \geq u_i(Y)$ where for one user a strict inequality holds. This means that $\langle Y' \rangle \succ \langle Y \rangle$ and, consequently, that Y is not LMMF, which contradicts the assumption. □

Proposition 4. *LMMF violates envy freeness.*

Proof. Consider the following example:
Example 3. A resource pool of a single resource $C = (1)$ and two users, each with single demand vector: $D_1 = \{(1)\}$ and $D_2 = \{(0.5)\}$.

The allocation under LMMF is $Y_1 = \frac{2}{3}$ and $Y_2 = \frac{1}{3}$ as both users get a utility of $\frac{2}{3}$. In this scenario, user 2 prefers the allocation of user 1 over his own, and hence LMMF is not EF. □

NB. We now prove that NB is both envy free and Pareto optimal.

Proposition 5. *NB is envy free and Pareto optimal.*

Proof. The Competitive Equilibrium from Equal Income (CEEI) [40] in economic theory is the allocation reached in a competitive market with multiple resources when each agent starts with an "endowment" of $\frac{1}{n}$ of each resource and then trades resources with the others. To prove the proposition we show that the allocation of NB coincides with that of the CEEI. To establish this, it suffices to show that the utility function of each user j in our setting is homogeneous, in the sense that $\gamma u_j(X) = u_j(\gamma X)$ for every scalar γ and vector of resources $X = (X_1, ..., X_k)$ (see [38], [40]). To see this, note that by the definition of a utility function in our model

$$u_j(\gamma X) = \max \sum_{m=1}^{M_j} \alpha_m \tag{19}$$

$$\text{Subject to} \quad \sum_{m=1}^{M_j} \alpha_m d_{jm} \leq \gamma X \tag{20}$$

$$\alpha_m \geq 0 \quad \forall m \in [M_j]$$

Constraints 20 can be rewritten as $\sum_{m=1}^{M} \frac{\alpha_m d_{jm}}{\gamma} \leq X$. Hence we can define $\alpha'_m \triangleq \frac{\alpha_m}{\gamma}$ and constraint (20) can be written as $\sum_{m=1}^{M} \alpha'_m d_{jm} \leq X$. Since $\max \sum_{m=1}^{M} \alpha_m = \gamma \max \sum_{m=1}^{M} \alpha'_m$ we get $\gamma u_j(X) = u_j(\gamma X)$. Hence, the allocation returned by NB is equivalent to the outcome of CEEI. CEEI is known to be PO and EF [39] and so the proof immediately follows. □

We point out that NB can easily be seen to not be lexicographically max-min fair. In fact, even if there is a single resource, NB will always split that resource equally disregarding how much it is worth to each user as the utility function is homogeneous.

C. Incentives

Proposition 6. *LMMF is not strategyproof.*

Proof. Consider Example 3. User 2 benefits by reporting $D'_2 = \{(1)\}$, since then the resource is divided equally. □

Proposition 7. *NB is not strategyproof.*

Proof. Consider the following example: A resource pool of $C = (1, 1)$ and two users with single demand vector each" $D_1 = \{(\frac{2}{3}, \frac{1}{3})\}$ and $D_2 = \{(\frac{1}{4}, \frac{3}{4})\}$. NB outputs $Y_1 = (\frac{12}{15}, \frac{6}{15}), Y_2 = (\frac{3}{15}, \frac{9}{15})$ and the corresponding utilities are $u_1(Y_1) = 1.2$ and $u_2(Y_2) = 0.8$. If user 2 reports $D'_2 = \{(\frac{1}{3}, \frac{2}{3})\}$ then NB outputs $Y_1 = (\frac{2}{3}, \frac{1}{3}), Y_2 = (\frac{1}{3}, \frac{2}{3})$ and the corresponding utilities are $u_1(Y_1) = 1$ and $u_2(Y_2) = 0.83$ (note that since user 2 misreports, not all of his obtained resources are in use). Hence, by misreporting, user 2 can improve his utility from 0.8 to 0.83. □

Proposition 8. *LMMF violates sharing-incentive.*

Proof. Consider resource pool $C = (1, 1, 1)$ and two users with demands $D_1 = \{(0, 1, 0)\}, D_2 = \{(1, 0, 0), (0, \frac{1}{2}, \frac{1}{2})\}$, where each user has "endowment" $E = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Note that $u_2(E) = \frac{3}{2}$. Then, the allocation under LMMF are given by $Y_1 = (0, 1, 0), Y_2 = (1, 0, 0)$. Hence, $u_2(Y_2) = \frac{1}{2}$ and then $u_2(E) > u_2(Y_2)$. □

Proposition 9. *NB satisfies sharing-incentive.*

Proof. As shown in Lemma 5, the output of NB is equivalent to CEEI. As CEEI satisfies sharing incentive, the proof follows. □

VI. Results for Linear Tradeoffs

As seen in the previous section, the LMMF mechanism does not guarantee many desired properties. We now show that in the very restricted setting in which tradeoffs are linear, that is, $\sum_r d^r_{jm} = 1$ for each user $j \in N$ and demand $d_{jm} \in D_j$, LMMF is, in fact, both envy free (EF) and strategyproof (SP). The following table summarizes our results for LMMF and NB for this class of so called "simplex demands". We regard these results as a first step and leave the analysis of other restricted classes of demands of interested to future research.

Properites	CE	EF	SI	PO	LMMF	SP	GSP
LMMF	✓	✓		✓	✓	✓	✓
NB	✓	✓	✓	✓			

We next present our proofs that LMMF is both EF and SP. The following lemma plays an important role in our proofs.

Lemma 6. *Let D be a specification of all users' resource-demands and u be the utility function over demands in D. Similarly, let D' be a different specification of all users' resource-demands, and u' be a utility function over demands in D'. Let $Y = (Y_1, \ldots, Y_n)$ be the output of the LMMF mechanism for D. Then, $u_i(Y) \geq u'_i(Y)$.*

Proof. Since the allocation Y_i is a convex combination of the resource demand vectors, the "packing coefficients" of u_i, i.e., $\alpha_1, ..., \alpha_{M_i}$, satisfies

$$\sum_{m=1}^{M_i} \alpha_m d^r_{i_m} = Y^r_i, \quad r \in [k] \tag{21}$$

Since D' is an arbitrary demand set, Y_i is not necessarily a combination of demand vector in D'_i. Thus, the "packing coefficients" of u', i.e., $\alpha'_1, ..., \alpha'_{M'_i}$, satisfies,

$$\sum_{m=1}^{M'_i} \alpha'_m d'^r_{i_m} \leq Y^r_i, \quad \forall r \in [k] \tag{22}$$

Summing the constraints (21) and (22) over all resources,

$$\sum_{r\in[k]} \sum_{m=1}^{M'_i} \alpha'_m d'^r_{i_m} \leq \sum_{r\in[k]} \sum_{m=1}^{M_i} \alpha_m d^r_{i_m} \tag{23}$$

Changing the order of summation and using the property that the demand vectors are over the simplex,

$$\sum_{m=1}^{M'_i} \alpha'_m \leq \sum_{m=1}^{M_i} \alpha_m$$

By definition of utility , $u_i(Y) \geq u'_i(Y)$. □

We are now ready to prove SP.

Theorem 3. *LMMF mechanism over simplex demands is strategyproof.*

Proof. Let i be a manipulative user and suppose by contradiction that i benefits by misreporting D'_i instead of reporting its real demand D_i. Let $D' = (D'_i, D_{-i})$ be the report of all users under the manipulation of i, and let Y' be the resulting allocation of LMMF under D'. Let Y be the resulting allocation of LMMF given that all users report their true demands (under D). We first show that if i benefits by lying then $i \in \theta(Y, Y')$; namely i is a pivot user. Let $q \in \theta(Y, Y')$ and $q' \in \theta(Y', Y)$ be pivot users. Suppose, for point of contradiction, that $u_q(Y_q) < u_i(Y)$. We now analyze three possibilities and show that under at each, there exists an allocation that contradicts the property of LMMF (see figure 6):

1) $u_q(Y) < u_{q'}(Y')$. Then, followed by Claim 2, $\langle Y' \rangle \succ \langle Y \rangle$.
2) $u_q(Y) = u_{q'}(Y')$. First note that it must be that $q \neq q'$. as otherwise it contradict the fact that q and q' are pivots. We can also derive by the definitions of the pivots q and q' that $u_q(Y') \geq u_{q'}(Y)$. Since $u_q(Y) = u_{q'}(Y')$ we get $u_q(Y') > u_q(Y)$. Followed by corollary 2, we get a contradiction to LMMF.
3) $u_q(Y) > u_{q'}(Y')$. To show a contradiction to LMMF, we define a dual scenario where the demand set of the users is D' instead of D, i.e., i's real demand is D'_i and the manipulation is D_i. Thus, in this case Y' is the truthful allocation (under D'). Let u' be the dual utility function, namely u' is utility function over demand D'. Since all users except for i report their true demands, their utility under the same allocation over demand sets D and D' is the same. Formally, for each user $j \neq i$

$$u'_j(Y') = u_j(Y') \text{ and } u'_j(Y) = u_j(Y) \tag{24}$$

Thus, we get a 'mirror image' of the utilities of the users, except for i. Note that without the presence of i, we had $q' \in \theta'(Y', Y)$, and hence, $u'_{q'}(Y) > u'_{q'}(Y')$, which by corollary 2, contradicts the assumption that Y' is a LMMF over D'. We need to show that q' is the pivot (with the presence of i). Indeed, $q' \in \theta'(Y', Y)$ since $u'_i(Y') > u'_{q'}Y'$, which followed by,

$$u'_i(Y') \geq u_i(Y') > u_i(Y) > u_q(Y) > u_{q'}(Y) = u'_{q'}(Y')$$

where the left inequality follows from Lemma 6, the second left inequality uses the assumption of manipulator, the third left inequality uses the case condition, the second right inequality follows as i is not a pivot, and the right equality follows (24) (this can be pictures in case 3 of figure 6).

Thus we get that the manipulative user i has to be the pivot (i.e, $i \in \theta(Y, Y')$). By assumption of the manipulative user $u_i(Y') > u_i(Y)$. But then, by corollary 2, Y is not LMMF. Thus, if i benefits by misreporting its demand, then this leads a contradiction to LMMF. Hence, LMMF mechanism over simplex demands is strategyproof. □

We note here that we can strengthen Theorem 3, showing that LMMF under simplex demand is *collusionproof* (CP). This can be proved using very similar argument to the proof of Theorem 3, showing that if there exists some manipulator user that improve its utility by misreporting, there must exists another manipulator user that is worse off if misreports its demand vectors.

Theorem 4. *LMMF is envy-free over simplex demands.*

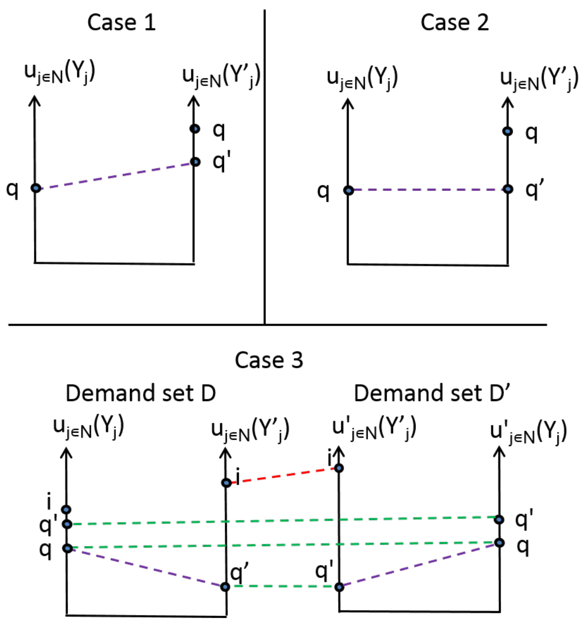

Fig. 6. Illustration of the cases in the proof of Theorem 3: in each case there is a pivot user (q in cases 1 and 2, and q' in case 3) that has higher utility in another allocation (which implies that Y is not LMMF under demand set D, or Y' is not LMMF under demand set D'). Case 3 analyzes the dual scenario under demand set D' (which is a mirror image of the original case, as shown by the green dashed line, except for user i). By Lemma 6 we have $u'_i(Y') \geq u_i(Y')$ (as shown by the red dashed line).

Proof. Let Y be an LMMF allocation. Suppose, by contradiction, that LMMF mechanism is not envy-free. This implies that there is a pair of user i, j such that $u_i(Y_j) > u_i(Y_i)$.There are two complementary cases:

1) Suppose that $u_i(Y_i) \geq u_j(Y_j)$. By Lemma 6 we get $u_j(Y_j) \geq u_i(Y_j)$ (as i's demand set is different from j's). Therefore, we have $u_i(Y_i) \geq u_i(Y_j)$. Contradiction to the assumption.
2) Suppose that $u_i(Y_i) < u_j(Y_j)$. In this case we argue that $u_i(Y_j) = 0$. Suppose by contradiction that $u_i(Y_j) > 0$. Let $\delta > 0$ be a (small) constant to be determined later. Suppose that user j transfers the vector $\delta \cdot Y_j$ to user i. Then the utility of j is $u_j(Y_j - \delta Y_j)$ and the utility of user i becomes $u_i(Y_i + \delta Y_j)$. Since the utility function is continuous (Lemma 3), for sufficiently small $\delta > 0$ it holds that $u_j(Y_j - \delta Y_j) > u_i(Y_i)$. Since the utility function is concave (Lemma 4), it follows that $u_j(\delta Y_j) \geq \delta u_j(Y_j)$. Using the assumption that $u_i(Y_j) > 0$ it follows that $u_i(Y_i + \delta Y_j) > u_i(Y_i)$. Thus, the vector $(u_i(Y_i + \delta Y_j), u_j(Y_j - \delta Y_j))$ is lexicographically larger than $(u_i(Y_i), u_j(Y_j))$. Let Y' be the allocation after the transfer. Since $Y_l = Y'_l$ for every $l \neq i, j$), we get $\langle Y' \rangle \succ \langle Y \rangle$. This is a contradiction to LMMF. Hence, $u_i(Y_j) = 0$. But then this implies that $u_i(Y_j) \leq u_i(Y_i)$ (as the utilities are non negative) and therefore contradicts the assumption.

□

LMMF is not SI under simplex demands, as the counter example in section V considers users with simplex demands.

VII. Extensions

We now briefly discuss extensions of the results presented in previous sections. Recall that the utility function considered in our model (Section III) reflects the number of resource-demands that can be packed in a given set of resources. Our results for LMMF actually extend to utility functions of the form

$$
\begin{aligned}
u_j(X) = \quad & \max_{\alpha_1, \alpha_2, \ldots, \alpha_{M_j}} f(\sum_{m=1}^{M_j} \alpha_m d_{jm}) \\
s.t. \quad & \sum_{m=1}^{M_j} \alpha_m d_{jm} \leq X \\
& \alpha_m \geq 0 \quad \forall m \in [M_j]
\end{aligned}
$$

where f is a concave function and satisfies strict monotonicity (i.e., for any positive γ, $f((1+\gamma)x) > f(x)$). In fact, our results for LMMF hold even if each user has a different function f, so long as these functions are publicly known. Lastly, when f is strictly concave, the LMMF outcome is guaranteed to be unique.

VIII. Conclusion

We initiated the study of fair resource-allocation with resource tradeoffs. We leave the reader with two interesting research directions: (1) We proposed and theoretically analyzed two mechanisms: LMMF and NB. Examining other mechanisms and, in particular, mechanisms that reflect other economic approaches to fair resource allocation is an interesting research direction; (2) We considered two extreme environments, namely, unrestricted tradeoffs and linear tradeoffs. Exploring other resource tradeoff settings on this spectrum is another interesting research direction. In particular, empirical studies of actual resource tradeoffs in cloud computing might motivate new questions along the lines outlined in this paper.

References

[1] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types." in *NSDI*, vol. 11, 2011, pp. 24–24.

[2] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multiresource allocation: fairness-efficiency tradeoffs in a unifying framework," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 6, pp. 1785–1798, 2013.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.

[4] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 242–253.

[5] S. J. Brams and A. D. Taylor, *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[6] J. B. Barbanel, A. D. Taylor *et al.*, *The geometry of efficient fair division*. Cambridge University Press Cambridge, 2005.

[7] S. K. Baruah, J. E. Gehrke, and C. G. Plaxton, "Fast scheduling of periodic tasks on multiple resources," in *Parallel Processing Symposium, International*. IEEE Computer Society, 1995, pp. 280–280.

[8] D. Zhu, D. Mossé, and R. Melhem, "Multiple-resource periodic scheduling problem: how much fairness is necessary?" in *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*. IEEE, 2003, pp. 142–151.

[9] Y. Liu and E. Knightly, "Opportunistic fair scheduling over multiple wireless channels," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2. IEEE, 2003, pp. 1106–1115.

[10] J. Kleinberg, Y. Rabani, and É. Tardos, "Fairness in routing and load balancing," in *Foundations of Computer Science, 1999. 40th Annual Symposium on*. IEEE, 1999, pp. 568–578.

[11] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking (ToN)*, vol. 8, no. 5, pp. 556–567, 2000.

[12] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, pp. 237–252, 1998.

[13] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: fair scheduling for distributed computing clusters," in *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*. ACM, 2009, pp. 261–276.

[14] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling," in *Proceedings of the 5th European conference on Computer systems*. ACM, 2010, pp. 265–278.

[15] D. C. Parkes, A. D. Procaccia, and N. Shah, "Beyond dominant resource fairness: extensions, limitations, and indivisibilities," in *Proceedings of the 13th ACM Conference on Electronic Commerce*. ACM, 2012, pp. 808–825.

[16] D. Dolev, D. G. Feitelson, J. Y. Halpern, R. Kupferman, and N. Linial, "No justified complaints: On fair sharing of multiple resources," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 2012, pp. 68–75.

[17] A. Gutman and N. Nisan, "Fair allocation without trade," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 719–728.

[18] W. Wang, B. Li, and B. Liang, "Dominant resource fairness in cloud computing systems with heterogeneous servers," *arXiv preprint arXiv:1308.0083*, 2013.

[19] D. C. Parkes, A. D. Procaccia, and N. Shah, "Beyond dominant resource fairness: Extensions, limitations, and indivisibilities," *ACM Transactions on Economics and Computation (TEAC)*, vol. 3, no. 1, p. 3, 2015.

[20] M. Chowdhury, Z. Liu, A. Ghodsi, and I. Stoica, "Multi-resource fairness for correlated and elastic demands," in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, 2016, pp. 407–424.

[21] S. M. Zahedi and B. C. Lee, "Sharing incentives and fair division for multiprocessors," *IEEE Micro*, 2015.

[22] T. Bonald and J. Roberts, "Multi-resource fairness: Objectives, algorithms and performance," in *ACM SIGMETRICS Performance Evaluation Review*. ACM, 2015.

[23] T. Lan, D. Kao, M. Chiang, and A. Sabharwal, *An axiomatic theory of fairness in network resource allocation*. IEEE, 2010.

[24] W. Wang, B. Li, B. Liang, and J. Li, "Multi-resource fair sharing for datacenter jobs with placement constraints," in *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2016.

[25] E. Meskar and B. Liang, "Fair multi-resource allocation with external resource for mobile edge computing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018.

[26] R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella, "Multi-resource packing for cluster schedulers," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 455–466.

[27] E. J. Friedman, V. Gkatzelis, C.-A. Psomas, and S. Shenker, "Fair and efficient memory sharing: Confronting free riders," in *Conference on Artificial Intelligence (AAAI)*, 2019.

[28] A. D. Procaccia and J. Wang, "Fair enough: Guaranteeing approximate maximin shares," in *Proceedings of the fifteenth ACM conference on Economics and computation*. ACM, 2014, pp. 675–692.

[29] D. Kurokawa, A. D. Procaccia, and J. Wang, "When can the maximin share guarantee be guaranteed?" in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[30] D. T. Nguyen, L. B. Le, and V. K. Bhargava, "A market-based framework for multi-resource allocation in fog computing," *IEEE/ACM Transactions on Networking*, 2019.

[31] S. M. Zahedi and B. C. Lee, "Ref: Resource elasticity fairness with sharing incentives for multiprocessors," in *ACM SIGPLAN Notices*. ACM, 2014.

[32] J. Khamse-Ashari, I. Lambadaris, G. Kesidis, B. Urgaonkar, and Y. Zhao, "An efficient and fair multi-resource allocation mechanism for heterogeneous servers," *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[33] J. Rawls, "Some reasons for the maximin criterion," *The American Economic Review*, 1974.

[34] C. A. Waldspurger and W. E. Weihl, "Lottery scheduling: Flexible proportional-share resource management," in *Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation*. USENIX Association, 1994, p. 1.

[35] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 4. ACM, 1989, pp. 1–12.

[36] B. Radunovic and J.-Y. L. Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 5, pp. 1073–1083, 2007.

[37] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE'08*. Ieee, 2008, pp. 1–10.

[38] J. Nash, "The bargaining problem," *Econometrica: Journal of the Econometric Society*, 1950.

[39] H. Varian, "Equity, envy, and efficiency," *Journal of economic theory*, 1974.

[40] H. Moulin, *Fair division and collective welfare*. MIT press, 2004.

[41] R. Cole, V. Gkatzelis, and G. Goel, "Mechanism design for fair division: allocating divisible items without payments," in *Proceedings of the fourteenth ACM conference on Electronic commerce*. ACM, 2013, pp. 251–268.

[42] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang, "The unreasonable fairness of maximum nash welfare," in *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 2016, pp. 305–322.

[43] D. Zarchy, D. Hay, and M. Schapira, "Capturing resource tradeoffs in fair multi-resource allocation," in *2015 IEEE Conference on Computer Communications, INFOCOM 2015, Kowloon, Hong Kong, April 26 - May 1, 2015*, 2015, pp. 1062–1070. [Online]. Available: https://doi.org/10.1109/INFOCOM.2015.7218479