Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Journal of Economic Theory 118 (2004) 209–228
JOURNAL OF Economic Theory
http://www.elsevier.com/locate/jet

Fair imposition

Ryan Porter,[a] Yoav Shoham,[a,*] and Moshe Tennenholtz[b]

[a] Computer Science Department, Stanford University, 353 Serra Mall, Gates Building Room 140, Stanford, CA 94305-9010, USA
[b] Industrial Engineering and Management, Technion-Israel Institute of Technology, Israel

Abstract

We introduce a new mechanism-design problem called *fair imposition*. In this setting a center wishes to fairly allocate tasks among a set of agents whose cost structures are known only to them, and thus will not reveal their true costs without appropriate incentives. The center, with the power to impose arbitrary tasks and payments on the agents, has the additional goal that his net payment to these agents is never positive (or, that it is tightly bounded if a loss is unavoidable). We consider two different notions of fairness that the center may wish to achieve. The central notion, which we call *k-fairness*, is in the spirit of max–min fairness. We present both positive results (in the form of concrete mechanisms) and negative results (in the form of impossibility theorems) concerning these criteria. We also briefly discuss an alternative, more traditional interpretation of our setting and results, in the context of auctions.

JEL classification: C7; D63

Keywords: Game theory; Mechanism design; Fairness

1. Introduction

Central allocation of tasks among multiple agents is a fundamental problem in several fields, including economics, computer science, and operations research. In this class of problems there is a center, or procurer, who aims to allocate one or more tasks among several agents (e.g., companies, employees, or computers) in a way that meets some set of criteria. In the setting we consider, *fair imposition*, the center possesses the ability to impose arbitrary behaviors on the individual agents, but has

*Corresponding author. Fax: +1-650-725-1449.
E-mail address: shoham@stanford.edu (Y. Shoham).

0022-0531/$ - see front matter
doi:10.1016/j.jet.2003.11.005

no access to their private information (specifically, their costs for performing the tasks). We assume that this dictator is frugal in that he wishes to relegate the entire cost to the agents (a goal formally referred to as *no deficit*). However, we also assume that the dictator is benign in the sense that he wishes the agent costs to be both equitable and as low as possible. Later we will formally define notions of fairness that capture these general goals.

The setting of fair imposition is quite natural. It can serve as a model of how a large corporation might divide an unexpected task among its business units. For example, consider a corporation with three factories, each operating as a profit center. Imagine they are all due for a centrally financed upgrade that would result in a boost to productivity, except that the magnitude of each boost is known only by the individual factory manager. Suppose that budget cuts will force the CFO to cancel one of the three factory upgrades, and he has the following considerations: he wishes the overall productivity of the firm to decline minimally, and he wishes that each of the managers feel that the outcome is fair. What protocol will achieve these goals?

It is not hard to come up with other applications of this general framework. For example, it can serve as a model of planning in a centralized economy.[1] The application that in fact served as initial motivation for this work is military air transportation. The US military uses civilian aircraft for a surprisingly high fraction of its transportation needs. This is true even in times of peace, and certainly in war times. In principle, the government has the ability to commandeer the aircraft at will, but of course that is not a tenable course of action. Instead it pays the civilian carriers, but the way in which it currently does so is largely ad hoc and inefficient. The models and protocols presented in this paper may, at least in principle, offer a better alternative.

What are the tools available to the center? In principle, all he can do is institute a *procurement protocol*, which is an orderly procedure of information exchange and an outcome function. The outcome function selects agent(s) to provide the service(s) and determines payments to or from the agents—both as a function of this information exchange. The trick is to design the protocol so as to induce the agents to exchange information in way that leads to an outcome desired by the center.

To show the issues that arise in the design of such a protocol, consider the following straw-man protocol for allocating a single task to one of n agents. Let $v_{[1]}$ be the lowest cost for this task among the agents. In this protocol the center asks each agent to declare its cost, assigns the task to the agent with the lowest declared cost (call it $\hat{v}_{[1]}$), pays $\frac{n-1}{n}\hat{v}_{[1]}$ to this agent, and collects a payment of $\frac{\hat{v}_{[1]}}{n}$ from each other agent. The net payments by the center are exactly zero, and if $\hat{v} = v$ then all agents suffer a loss of exactly $\frac{v_{[1]}}{n}$. Notice that this outcome minimizes inequity among the agents and bounds the loss of each agent at the lowest possible amount, given our requirement of no deficit. Formally, we define 1-fairness to be achieved if no agent suffers a loss of more than $\frac{v_{[1]}}{n}$. More generally, k-fairness is achieved when no agent

[1] We thank John McMillan for this observation.

loses more than $\frac{v_{[k]}}{n}$, where $v_{[k]}$ is the k-lowest cost among the agents. Notice that this notion fairness is based on maximin principle of [8]. The comparable idea in computer science is max–min fairness, which is a widely used criterion for fair division of bandwidth among a set of users (see, e.g., [1]). Also note that this mechanism could not have achieved 1-fairness (in conjunction with no deficit) if it had not assigned the task to the agent with the lowest cost. We will show a general result that we get economic efficiency "for free"; that is, given our other desiderata, we can assume without loss of generality that it is satisfied.

However, this protocol has a crucial flaw: the assumption that each agent will truthfully reveal its cost is obviously invalid, because the agent who submits the lowest cost has an incentive to inflate its declared cost to an amount just below the second lowest declared cost. By increasing $\hat{v}_{[1]}$ in this way, this agent would increase its payment from the center.

This example illustrates why the center must resort to incentive engineering of the sort encountered in mechanism design. Indeed, the reader familiar with mechanism design (for an introduction see [6]) might be tempted to view fair imposition as already addressed by that literature. However, the technical differences are substantive, and the solutions called for are different as a result. Although we start the formal treatment only in the next section, for such readers let us make the following technical comment: in the setting of fair imposition we retain the requirements of incentive compatibility, no deficit, and economic efficiency, jettison the requirement of individual rationality, and add a new requirement of fairness.

Although we feel that the setting as discussed is natural and important, for the same class of expert readers we should also mention an alternative interpretation of our framework and results. This is in the more standard context of auctions, where the mechanism designer wishes to not only maximize social welfare, but to also share the surplus among the agents as evenly as possible via side payments.[2] We will discuss this interpretation only briefly, and in two places—after the technical exposition, and in the conclusions section.

The rest of the paper is organized as follows. In Section 2 we formally define the single-task fair imposition setting as a mechanism design problem (including the alternative interpretation of our setting in the context of an auction). Section 3 presents our basic results concerning the feasibility of fair allocation: when no deficit is a strict requirement, we prove that we cannot achieve 2-fairness (and thus also not 1-fairness) and present a mechanism that achieves 3-fairness (and thus k-fairness for $k > 3$ as well). An inequity of the mechanism used for this possibility result, in which one agent pays more than other agents despite having a lower cost, leads into a discussion and an impossibility result concerning our second type of fairness—avoiding what we will call a *competence penalty*. We also give a mechanism that achieves 1-fairness while incurring a minimal deficit. The techniques and results presented for the single task case can be generalized to multiple tasks. We demonstrate this possibility in Section 4 by extending the setting to two (possibly

[2] We thank an anonymous reviewer for pointing out this connection.

interacting) tasks and providing similar mechanisms and impossibility results. Concluding remarks are given in Section 5.

Note that, for purposes of readability, full proofs are postponed to the appendix and replaced in the text with proof sketches.

2. Setting

We construct the fair imposition setting for the single task case within the basic mechanism design framework. While most of our criteria are consistent with those of mechanism design, the novelty of our setting lies in our new goal of k-fairness.

2.1. The mechanism design problem

There exists a single task, a center, and a set of agents $N = \{1, 2, \dots, n\}$ who can accomplish the task. Each agent i has a privately known type $v_i \in \Re_+$, which represents its non-negative cost to execute the task. The center assigns the task to an agent and collects a payment (which can be negative) from each agent through the use of a mechanism $\Gamma = \langle B, f(\cdot) \rangle$.

The set of possible strategies for each agent i is defined by $B = \{b_i \mid b_i : \Re_+ \to \Re_+\}$. A strategy b_i for agent i maps each possible type to a declared type, $b_i(v_i)$, which will also be referred to as $\hat{v}_i$ (or as v_i if $\hat{v}_i = v_i$). By restricting the space of messages for an agent to declared types, we are only considering *direct mechanisms*.

The function $f : \Re_+^n \to O$ takes as input a declared type from each agent and returns an outcome in the set O. An outcome $o \in O$ is a pair of vectors (g, t), where $g = (g_1, \dots, g_n)$ and $t = (t_1, \dots, t_n)$. Each $g_i \in \{0, 1\}$ represents whether or not agent i is assigned the task, and each $t_i \in \Re$ is the amount that agent i must pay to the center.[3] For simplicity, we will use $g_i(v)$ and $t_i(v)$ to represent the corresponding terms in the outcome $f(v)$. We also have the restriction that $\sum_{i=1}^n g_i = 1$ in order to capture the fact that the task can only be assigned to one agent. We will overload the function $f(\cdot)$ so that it can take as its arguments the declared types of $n-1$ agents, instead of the standard n. In this case, the mechanism is restricted so that $g_j = 0$ and $t_j = 0$ for the agent j whose type was not given as an argument. While the set of possible strategies B is constant for all mechanisms, $f(\cdot)$ is the degree of freedom used to satisfy the goals of the mechanism designer.

Each agent i has a linear utility function that depends on both the outcome and the agent's own type: $u_i((g, t), v_i) = -g_i \cdot v_i - t_i$. We assume that all agents are rational, in the sense that they are expected-utility maximizers.

In the sequel we will use the notational shorthand $v = (v_1, v_2, \dots, v_n)$ for the vector of types of all agents. The vector of all types excluding that of agent i is $v_{-i} = (v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n)$. We can then refer to v as (v_i, v_{-i}). For a vector v, we denote

[3] We could allow g to be any value between 0 and 1 to denote the probability that agent i is selected as the service provider. However, we will later show that it suffices to only consider mechanisms that always assign the task to the assign with the lowest cost.

the jth lowest cost by $v_{[j]}$. In this way, we can use $v_{-i_{[j]}}$ to represent the jth lowest cost among all agents other than agent i. Similarly, we will use $b = (b_1, \ldots, b_n)$ to represent a vector of strategies for all agents. Then, $b(v)$ would be the vector $(b_1(v_1), \ldots, b_n(v_n))$. Without loss of generality, we will assume that the agents are sorted in non-decreasing order of cost. That is, $v_1 \leqslant v_2 \leqslant \cdots \leqslant v_n$. This assumption is solely for expositional purposes—the center has no knowledge of any such ordering.

2.2. Mechanism criteria

Four requirements that are often present in a mechanism-design setting are incentive compatibility, no deficit, economic efficiency, and individual rationality. We retain the first three as requirements. The fourth, individual rationality, which requires that all agents always have non-negative utility for the outcome of the mechanism, is often present when agents are assumed to have the option of not participating in the mechanism. Since this assumption does not hold in our setting, where the center can force the agents to both provide a desired service and make payments to the center, we replace individual rationality with the notion of k-fairness. In this section we formally define each of our criteria.

We say that incentive compatibility holds when each agent maximizes its utility by declaring its true type, regardless of the declarations of all other agents.

Definition 1. A mechanism satisfies *incentive compatibility* (IC) if for all i and $v_i, u_i(f(v_i, \hat{v}_{-i}), v_i) \geqslant u_i(f(v_i', \hat{v}_{-i}), v_i)$ holds for all v_i', and $\hat{v}_{-i}$.

Our second requirement, no deficit, requires that the center never lose money.

Definition 2. A mechanism satisfies *no deficit* (ND) if for all $\hat{v}: \sum_i t_i(\hat{v}) \geqslant 0$.

A third desideratum is economic efficiency, which simply requires that in every equilibrium, our choice rule $g(\cdot)$ select the lowest cost agent as the service provider, with ties broken arbitrarily.

Definition 3. A mechanism satisfies *economic efficiency* (EE) if for all v and all equilibria b, there exists an agent j such that: $g_j(b(v)) = 1$ and $v_j = v_{[1]}$.

We now define the notion of fairness motivated in the introduction. In words, k-fairness holds if in all equilibria the utility loss of each agent is bounded by kth lowest cost among the agents, divided by the number of participating agents.

Definition 4. For any $1 \leqslant k \leqslant n$, a mechanism satisfies *k-fairness* if for all v and all equilibria b, $u_i(f(b(v)), v_i) \geqslant -\frac{v_{[k]}}{n}$ holds for each agent i.

Based on the preceding two definitions, we can see the need for incentive compatibility, because both EE and k-fairness depend on the true type of the agents, despite the fact that the mechanism can only base g and t on the declared types. If IC is satisfied, then we only have to consider the single equilibrium in which all agents declare their type truthfully (that is, $b(v) = v$). On the other hand, if IC is not satisfied, then both EE and k-fairness, as defined here, are unreasonable goals for a mechanism. Thus, in the sequel they both will only appear as goals of a mechanism in conjunction with the goal of IC. While an indirect mechanism that merely induces a dominant strategy equilibrium (with corresponding changes to the definition of EE and k-fairness) would have sufficed, the Revelation Principle for Dominant Strategies (see, for example, [6]) tells us that, without loss of generality, we can restrict our space of mechanisms to those in which truthful revelation of types is a dominant strategy.

2.3. *Alternative interpretation*

While in our motivating examples each v_i would be positive, since it represents the cost of completing the task, our formal setting does not restrict v_i in any way. An alternative interpretation of our formulation, which makes a more apparent connection to the existing mechanism design literature, is as a private-value auction for an indivisible good. In this case, each v_i would be negative, representing the value (or negative cost) of the good to the particular agent.

While the mechanisms we present would satisfy individual rationality when each v_i is negative, they would instead suffer from a free-rider problem, in which agents who have no value for the good would have incentive to participate because they would receive a positive payment from the center. Thus, our mechanisms apply best to settings in which the bidders all have an equal, a priori claim to the object (e.g., siblings at a probate court), and thus we wish to make the auction as "fair" as possible by having the winner compensate the losing bidders. In the conclusions section we will revisit our results in the context of this interpretation.

3. Results

We begin by proving that the requirement of EE will never prevent us from finding a mechanism that satisfies our other requirements. Then, we take ND as a firm requirement and show that we cannot achieve 2-fairness, but can achieve 3-fairness. The mechanism we construct for the possibility result has the property that an agent with a lower cost than other agents is forced to pay more than these agents. We then prove that this *competence penalty* is unavoidable. We conclude this section with a mechanism that achieves 1-fairness, at the cost of a slight relaxation of our ND requirement.

3.1. *Restrictions to EE mechanisms*

Before moving onto possibility and impossibility results, it will be helpful to restrict the space of mechanisms we need to consider. Specifically, if our goal is to find a mechanism that satisfies IC, k-fairness and ND, then we can limit our search to economically efficient mechanisms.

Lemma 1. *If there exists a mechanism that satisfies IC, k-fairness and ND, then there exists a mechanism that satisfies IC, k-fairness, ND, and EE.*

Proof (Sketch). By construction. Start with any mechanism Γ (defined by $g(\cdot)$ and $t(\cdot)$) that satisfies IC, k-fairness and ND. Considering each possible v separately, we transform Γ into a mechanism that maintains these properties and also satisfies EE. Let j be the agent who is the service provider selected by Γ (that is, $g_j(v) = 1$). If $j = 1$, then Γ already satisfies EE and we are done for this particular v (recall that we have ordered the agents by cost). Otherwise, we set $g_1(v) = 1$ and $g_j(v) = 0$. Then, to keep utility constant for all agents, we set $t_1(v) \leftarrow t_1(v) - v_1$ and $t_j(v) \leftarrow t_j(v) + v_j$. Constant utility means that IC and k-fairness continue to hold, and ND continues to hold because the center's net revenue from the agents changes by $(v_j - v_1) \geqslant 0$. $\square$

3.2. An impossibility result for 2-fairness

When we require IC and ND, it is not surprising that 1-fairness is not achievable. These three conditions together force the utility of each agent to be exactly $-\frac{v_{[1]}}{n}$. The fact that $v_{[1]}$ is unknown makes this impossible to accomplish. However, based on the success of the Groves mechanism (see [3]), one might expect that a "second-best" solution would be possible, allowing us to achieve 2-fairness. However, this is not the case.

Theorem 1. *There does not exist a mechanism that satisfies IC, 2-fairness, and ND, for any $n \geqslant 2$.*

Proof (Sketch). We will use a proof by contradiction that works for all $n \geqslant 2$. Assume that a mechanism does exist for a given $n \geqslant 2$ that satisfies IC, 2-fairness, and ND. Because of IC, we can assume that all agents declare truthfully, and Lemma 1 allows us to assume that a mechanism exists that also satisfies EE (that is, $g_1(v) = 1$).

A useful property (that we will use in later proof sketches) of mechanisms that satisfy k-fairness, IC, and EE is that they must pay each agent an amount that would satisfy k-fairness even if the agent's true type were at the boundary of changing which agent has the lowest cost. That is, agent 1 must be paid as if its type were v_2, and all other agents must be paid as if their type were v_1. The reason that this property holds is that IC demands that an agent's payment be constant for all declarations that do not change the service provider. Otherwise, there must be some vector v in which this agent has an incentive to lie, because the only other factor in the agent's utility function $(-g_i \cdot v_i)$ does not change. Furthermore, the requirement that the mechanism satisfy k-fairness places a bound on this payment. In the worst case (from the center's point of view), the service provider's cost is equal to the second-lowest cost (thus requiring the maximum amount of reimbursement for executing the task). For each of the non-service provider, the worst case is that their cost is equal to lowest cost (thus minimizing the kth lowest cost).

Applying this rule, it must be the case that the constant value that agent 1 pays is bounded by $t_1(v_1, v_{-1}) \leqslant -\frac{n-1}{n} v_2$. For the other agents ($i \neq 1$), the bound is $t_i(v_i, v_{-i}) \leqslant v_1/n$, because v_1 becomes the second lowest cost when this agent's type is considered to be v_1.

We can then show the following upper bound on the net payments to the center: $t_1(v) + \sum_{i \neq 1} t_i(v) \leqslant -\frac{n-1}{n} v_2 + (n-1) \cdot \frac{v_1}{n}$. Since $v_2 > v_1$ is possible, ND is not satisfied, reaching a contradiction. □

Because k-fairness is strictly more difficult to satisfy than $(k-1)$-fairness, we have the following corollary.

Corollary 1. *There does not exist a mechanism that satisfies IC, 1-fairness, and ND, for any $n \geqslant 2$.*

Note that we cannot show this result for $n = 1$. A counterexample is a mechanism that simply assigns the task to the single agent and pays it exactly zero.

3.3. A possibility result for 3-fairness

We now construct a mechanism to show that 3-fairness is the minimal level of fairness that we can achieve in conjunction with ND and IC.

Mechanism Fair3:

- Each agent i submits a declared cost $\hat{v}_i$.
- An agent j with the lowest declared cost is selected (that is, $\hat{v}_j = \hat{v}_{[1]}$).
- Assignment and payment rules are constructed as follows:
  - $\forall i \; g_i(\hat{v}) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases}$
  - $\forall i \; t_i(\hat{v}) = \begin{cases} \frac{\hat{v}_{-i_{[2]}}}{n} - \hat{v}_{[2]} & \text{if } i = j, \\ \frac{\hat{v}_{-i_{[2]}}}{n} & \text{otherwise.} \end{cases}$

In words, the service provider is reimbursed an amount ($\hat{v}_2$) such that this agent is the only one who can potentially make a profit from this "transaction". The rest of the payment rule $\left(\frac{\hat{v}_{-i_{[2]}}}{n}\right)$ for each agent i is equal to the second lowest declared cost among all agents other than agent i.

Theorem 2. *Mechanism Fair3 satisfies IC, 3-fairness, ND, and EE, for all $n \geqslant 3$.*

Proof (Sketch). IC holds because the payments fit the Groves scheme [3]. EE then follows from IC and the definition of Fair3. Each agent's payment consists of an "offset payment" $\left(\frac{\hat{v}_{-i_{[2]}}}{n}\right)$ that does not depend on the agent's type, plus the amount by which the agent's presence affects the costs incurred by the other agents. This amount is $-\hat{v}_2$ for the service provider, because without the service provider the agent with the second-lowest cost would be forced to complete the task, and zero for all other agents, because their presence obviously does not change which agent is selected as the service provider.

Given IC, the payments are: $t_1(v) = -v_2 + \frac{v_3}{n}$ for the service provider, $t_2(v) = \frac{v_3}{n}$ for agent 2, and $t_i(v) = \frac{v_2}{n}$ for all remaining agents ($i \geqslant 3$). Thus, ND holds because: $\sum_i t_i(v) = (-v_2 + \frac{v_3}{n}) + \frac{v_3}{n} + \sum_{i \geqslant 3} \frac{v_2}{n} \geqslant -v_2 + \frac{n}{n} v_2 = 0$.

Since IC holds, $u_i(f(v), v_i) = -t_i(f(v), v_i)$ holds for all non-service providers, and $u_1(f(v), v_1) = -v_1 - t_1(f(v), v_1) = -v_1 + v_2 - \frac{v_3}{n} \geqslant -\frac{v_3}{n}$ holds for the service provider, we can conclude that 3-fairness holds. □

3.4. Competence penalty

A closer look at mechanism *Fair3* reveals a disturbing inequity of payments among the agents. Notice that the agent with the second lowest cost pays $\frac{v_3}{n}$, while the other non-service providers pay $\frac{v_2}{n}$. The existence of inequity, in and of itself, is not very troublesome, and could even be justified if the agents who pay less had declared a lower cost, intuitively making themselves more "valuable" to the mechanism. However, *Fair3* produces the opposite effect. The agent who submitted the lowest valuation among the non-service providers pays the most of this group, enduring what we will call a *competence penalty*.

Definition 5. A mechanism enforces a *competence penalty* if there exists a vector v and two distinct agents i and j such that $g_i(v) = g_j(v) = 0$, $v_i < v_j$, and $t_i(v) > t_j(v)$.

Note that we have restricted the definition to only consider the non-service providers. We could have included the service provider into this definition, but our negative result below holds even for the current definition.

The *competence penalty* present in mechanism *Fair3* turns out to be surprisingly unavoidable. Not only is it impossible to construct a mechanism that satisfies 3-fairness and ND and is free of this type of inequity, we cannot even settle for n-fairness.

Theorem 3. *There does not exist a mechanism that satisfies IC, n-fairness, and ND, and that does not enforce a competence penalty, for any fixed number of agents $n \geqslant 2$.*

Proof (Sketch). The proof of this result for $n = 2$ follows directly from Theorem 1. For any $n \geqslant 3$, assume that a mechanism does exist that satisfies IC, n-fairness, and ND, and that does not enforce a competence penalty. By Lemma 1, we can assume that $g_1(v) = 1$.

The first step towards a contradiction is to prove by induction that for each agent i such that $1 < i \leqslant n$ (i.e., the non-service providers), $t_i(v) \leqslant v_{i-1}/n$ must hold. Starting with the base case of $i = n$, we must show that $t_n(v) \leqslant v_{n-1}/n$. Because of n-fairness, $t_n(v) \leqslant v_n/n$ must hold. Since the center must pay agent n as if its type were v_1 (using the argument presented in the proof sketch of Theorem 1), this bound becomes $t_n(v) \leqslant v_{n-1}/n$, because the nth lowest cost becomes the v_{n-1} when agent n's declaration is changed to v_1.

The inductive step for each i in the range $1 < i < n$ proceeds similarly. By the inductive assumption, $t_{i+1}(v) \leqslant v_i/n$. We need to show that $t_i(v) \leqslant v_{i-1}/n$. To avoid a *competence penalty*, $t_i(v) \leqslant t_{i+1}(v) \leqslant v_i/n$. The bound $t_i(v) \leqslant v_i/n$ becomes $t_i(v) \leqslant v_{i-1}/n$ by the same argument used for the base case.

We complete the proof by showing that these bounds prevent the mechanism from satisfying n-fairness, which we show here for the case of $n = 3$. Using the bounds we just derived, the center can collect a maximum of $\frac{v_1}{3} + \frac{v_2}{3}$ from agents 2 and 3. As shown in the proof sketch of Theorem 1, agent 1 must be paid $\frac{n-1}{n} v_2 = \frac{2}{3} v_2$. Since $v_1 < v_2$ is possible, ND is violated, a contradiction. □

Since $(k-1)$-fairness implies k-fairness, the following corollary trivially follows.

Corollary 2. *There does not exist a mechanism that satisfies IC, k-fairness, and ND, and does not enforce a competence penalty, for any* $1 \leqslant k < n$ *and any fixed number of agents* $n \geqslant 2$.

3.5. A possibility result for 1-fairness

Given the fact that ND forces us to settle for 3-fairness, a natural question is how much the center must pay in order to achieve 1-fairness. Indeed, while in some situations the center can expect to pay nothing, in many others it cannot demand as much. For example, in the military transportation domain which motivated this work, the relationship between the government and the airlines is not a simple one of dictator and subjects. If Fair3 were implemented, one would expect airlines to balk at the fact that the government not only receives the flight for free, but actually makes a profit. For these two reasons, we present a protocol that sacrifices ND in a minimal way in order to achieve 1-fairness.

Mechanism BoundedFair1:

- Each agent i submits a declared cost $\hat{v}_i$.
- An agent j with the lowest declared cost is selected (that is, $\hat{v}_j = \hat{v}_{[1]}$).
- Assignment and payment rules are constructed as follows:
  - $\forall i \;\; g_i(\hat{v}) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases}$
  - $\forall i \;\; t_i(\hat{v}) = \begin{cases} \frac{\hat{v}_{-i_{[1]}}}{n} - \hat{v}_{[2]} & \text{if } i = j, \\ \frac{\hat{v}_{-i_{[1]}}}{n} & \text{otherwise.} \end{cases}$

To quantify the amount of budget deficit the center suffers, we introduce the following definition.

Definition 6. A mechanism satisfies *m-bounded deficit* (m-BD) if for all $\hat{v}$, $\sum_i t_i(\hat{v}) \geqslant -m$.

We can show that the budget deficit for mechanism BoundedFair1 is capped by the difference in cost between the two lowest cost agents.

Theorem 4. *Mechanism* BoundedFair1 *satisfies IC*, 1-*fairness*, $(v_2 - v_1)$-*BD*, *and EE*, *for any* $n \geqslant 2$.

Proof (Sketch). IC follows in the same way as it did for Fair3, because the only change is in the "offset" component $\left(\frac{\hat{v}_{-i_{[1]}}}{n}\right)$ of the payment function: it now uses the *lowest* cost among the other agents rather than the second lowest cost. EE follows directly from IC and the definition of the mechanism. 1-fairness holds because of IC and the fact that the payment for the service provider is $t_1(v) = -v_2 + \frac{v_2}{n}$, causing its utility is then $u_1(f(v), v_1) = -v_1 - (-v_2 + \frac{v_2}{n}) = -\frac{v_1}{n} - \frac{n-1}{n}v_1 + \frac{n-1}{n}v_2 \geqslant -\frac{v_1}{n}$, while the utility for a non-service provider $(i > 1)$ is the negative of its transfer function: $u_i(f(v), v_i) = -\frac{v_1}{n}$. Finally, to show $(v_2 - v_1)$-BD: $\sum_i t_i(v) = (-v_2 + \frac{v_2}{n}) + \sum_{i \neq 1} \frac{v_1}{n} = -\frac{n-1}{n}v_2 + \frac{n-1}{n}v_1 \geqslant -(v_2 - v_1)$. □

The power of this protocol lies in the fact that it only requires two proficient agents in order to be reasonable for the center. One would expect $v_2 - v_1 \ll v_1$ to hold for a set of airlines when the task is a flight between two major cities. For example, suppose the two cheapest airlines could provide the flight for \$750K and \$800K. While the government would be unwilling to pay the full \$750K, a payment of \$800K – \$750K = \$50K may be acceptable. An additional advantage of this mechanism is that it does not enforce a *competence penalty*, because all non-service providers now pay the same amount.

4. Multiple task setting

So far, our study has concentrated on the imposition of a single task on a set of agents. If there are several independent tasks, then we can execute a separate protocol for each task and apply the techniques and results we previously obtained. However, it is often the case that the cost to complete a set of tasks is not simply the sum of the costs of the individual tasks. For example, a carrier's cost for a pair of flights might be lower than the sum of the costs for the individual flights when the destination of the first flight is the origin of the second. Alternatively, the cost for a pair could be higher when both flights originate from the same city and the airline only has one plane at this location. In such cases, the solution is not as simple, but we can still achieve results comparable to those of the single task setting. In this section we consider the case of two interacting services, but this number is only chosen for ease of exposition. The generalizations and techniques presented can be easily applied to any number of interacting tasks.

We first need to extend our formulation. In the new setting, which we call the *Multiple Task Setting* (MTS), there are two tasks, 1 and 2. We will use the variable s to index an individual task and S for a set of tasks. The type of agent i is now expanded to a function, $v_i : \{\{1\}, \{2\}, \{1,2\}\} \to \Re_+$, which maps each non-empty subset of the two tasks to a non-negative cost. The possible interaction between the services is reflected by the absence of a restriction that $v_i(\{1,2\}) = v_i(\{1\}) + v_i(\{2\})$. We also extend the notation for the ranking of a valuation so that $v_{[k]}(S)$ is the kth lowest cost to complete the tasks in S.

Additionally, g_i is expanded to a vector $(g_{i,1}, g_{i,2})$, where $g_{i,s} \in \{0,1\}$ denotes whether or not task s is assigned to agent i. The restriction that each task is only assigned once is now captured by: $\sum_{i=1}^{n} g_{i,s} = 1$ for $s = 1,2$.

For convenience we will add some new notation. The cost of an assignment g_i to agent i is represented by the function $c_i(g_i, v_i) = g_{i,1}(1 - g_{i,2}) \cdot v_i(\{1\}) + g_{i,2}(1 - g_{i,1}) \cdot v_i(\{2\}) + g_{i,1} \cdot g_{i,2} \cdot v_i(\{1,2\})$, where the terms of the form $(1 - g_{i,s})$ are used to avoid double counting the cost of completing a task. The utility function of agent i is then: $u_i(o, v_i) = -c_i(g_i, v_i) - t_i$. We will also use an aggregate cost term for the total cost of all agents: $c(g, v) = \sum_i c_i(g_i, v_i)$.

The definition of k-fairness can be extended in various ways for this setting. The extension we choose reflects a minimal change from our original definition: the bound on the loss of each agent for k-fairness in the current setting is the sum of the bounds for k-fairness of the single task setting when both of the tasks considered separately. This is an admittedly weak extension, and stronger extensions that consider the potential sub-additivity of costs do not allow results as strong as the ones we will show.

Definition 7. A mechanism satisfies *k-fairness* in the Multiple Task Setting if for all b and v, $u_i(f(b(v)), v_i) \geqslant -\frac{v_{[k]}(\{1\}) + v_{[k]}(\{2\})}{n}$ holds for each agent i.

Since the definitions of IC and ND carry over from the original setting, the only other definition we need to update is that of EE.

Definition 8. A mechanism satisfies *economic efficiency* (EE) in the Multiple Task Setting if for all b and v, $c(g(b(v)), v) \leqslant c(g'(b(v)), v)$ holds for all $g'(\cdot)$.

In words, there can be no assignment rule $g'(\cdot)$ that assigns tasks to agents in a way that reduces the total cost to the agents. Define $g^*(\cdot)$ to be any function $g(\cdot)$ that satisfies EE (with the tie-breaking rule being a degree of freedom).

4.1. MTS: an impossibility result for 2-fairness

It is easy to extend our infeasibility results for 2-fairness to the current setting. It is always possible that task 2 is a "dummy task" that all agents can complete at no cost (that is, $\forall i\ v_i(\{2\}) = 0$ and $v_i(\{1,2\}) = v_i(\{1\})$). In this case, the setting reduces to our original one.

Proposition 5. *There does not exist a mechanism that satisfies IC, 2-fairness, and ND, for any $n \geqslant 2$, in the Multiple Task Setting.*

4.2. MTS: a possibility result for 3-fairness

To achieve 3-fairness with our other requirements, we again borrow ideas from the Groves mechanism by aligning the interest of each individual agent with those of the entire system, while charging each agent an offset payment that does not depend on their own declared cost in order to satisfy ND. The "offset" payment is simply the sum of the two offset payments that would exist if Fair3 were executed twice. The rest of the payment rule for each agent i is the costs incurred by all other agents $j \neq i$ from execution of tasks when agent i participates minus the amount that these costs would be otherwise.

Mechanism Fair3b:

- Each agent i submits a declared cost $\hat{v}_i$.
- Assignment and payment rules are constructed as follows:
  - $\forall i, s\ g_{i,s}(\hat{v}) = g^*_{i,s}(\hat{v})$,
  - $\forall i\ t_i(\hat{v}) = \frac{\hat{v}_{-i_{[2]}}(\{1\}) + \hat{v}_{-i_{[2]}}(\{2\})}{n} + \sum_{j \neq i} c_j(g^*_j(\hat{v}), \hat{v}_j) - \sum_{j \neq i} c_j(g^*_j(\hat{v}_{-i}), \hat{v}_j)$.

We now show the result of executing mechanism Fair3b when the participating agents are the three from Table 1.

If $\hat{v} = v$, then task 1 is assigned to agent 3 and task 2 is assigned to agent 1 (that is, the only two non-zero values for $g(\cdot)$ are $g_{3,1}(v) = 1$ and $g_{1,2}(v) = 1$). The transfer functions are: $t_1(v) = \frac{14+16}{3} + 5 - (5 + 13) = -3$, $t_2(v) = \frac{11+16}{3} + (5 + 10) - (5 + 10) = 9$, and $t_3(v) = \frac{14+13}{3} + 10 - 20 = -1$.

If we modified the type of agent 2 so that $v_2(\{1,2\}) = 14$, then Fair3 would instead assign both tasks to this agent (that is, $g_{2,1}(v) = 1$ and $g_{2,2}(v) = 1$). The transfer functions would then be: $t_1(v) = \frac{14+16}{3} + 14 - 14 = 10$, $t_2(v) = \frac{11+16}{3} + 0 - (5 + 10) = -6$, and $t_3(v) = \frac{14+13}{3} + 14 - 14 = 9$.

Proposition 6. *Mechanism Fair3b satisfies IC, 3-fairness, ND, and EE, for any $n \geqslant 2$, in the Multiple Task Setting.*

Table 1
Types of agents used in example of Mechanism Fair3b

i	$v_i(\{1\})$	$v_i(\{2\})$	$v_i(\{1,2\})$
1	11	10	21
2	14	13	20
3	5	16	30

We omit the proof because of its similarity to that of both Fair3 and the Groves mechanism.

4.3. MTS: competence penalty

Given the similarity between mechanisms Fair3 and Fair3b, it is not surprising that Fair3b also enforces a competence penalty on the agents. Even if we extend the definition of competence penalty in a relatively weak manner, we can still show the same impossibility result as we did for the single task setting.

Definition 9. A mechanism enforces a *competence penalty* in the Multiple Task Setting if there exists a vector v and two distinct agents i and j such that the following conditions all hold:

(1) $g_{i,s}(v) = g_{j,s}(v) = 0$, for $s = 1, 2$,
(2) $\forall S \; v_i(S) \leqslant v_j(S)$,
(3) $\exists S \; v_i(S) < v_j(S)$,
(4) $t_i(v) > t_j(v)$.

Like the previous definition, this one only considers agents who were not assigned a task. In words, a mechanism enforces a competence penalty if one non-service provider pays more than another even though its type weakly "dominates" that of the other agent.

We can then show the following impossibility result.

Proposition 7. *There does not exist a mechanism that satisfies IC, n-fairness, and ND, and that does not enforce a competence penalty, for any fixed number of agents $n \geqslant 2$, in the Multiple Task Setting.*

The proof is omitted because of its similarity to that of the original impossibility result. Intuitively, we can again use the possibility of one of the tasks being a "dummy task".

4.4. MTS: a possibility result for 1-fairness

Additionally, we can extend the mechanism BoundedFair1 in order to achieve 1-fairness while accepting a (relatively small) budget deficit.

Mechanism BoundedFair1b:

- Each agent i submits a declared cost $\hat{v}_i$.
- Assignment and payment rules are constructed as follows:
  - $\forall i, s \; g_{i,s}(\hat{v}) = g^*_{i,s}(\hat{v})$,
  - $\forall i \;\; t_i(\hat{v}) = \frac{\hat{v}_{-i_{[1]}}(\{1\}) + \hat{v}_{-i_{[1]}}(\{2\})}{n} + \sum_{j \neq i} c_j(g^*_j(\hat{v}), \hat{v}_j) - \sum_{j \neq i} c_j(g^*_j(\hat{v}_{-i}), \hat{v}_j)$.

The bound on the budget deficit for this mechanism is a simple extension of the bound shown for the single task setting.

Proposition 8. *Mechanism* BoundedFair1b *satisfies IC*, 1-*fairness*, *EE*, *and* $[v_{[2]}(\{1\}) - v_{[1]}(\{1\}) + v_{[2]}(\{2\}) - v_{[1]}(\{2\})]$-*BD*, *for any* $n \geqslant 2$, *in the Multiple Task Setting*.

5. Conclusion

In this paper we investigated the addition of fairness as a goal in the mechanism-design setting. The novelty of this work lies not in the consideration of fairness—about which there is of course substantial literature in economics—but in its consideration in the context of mechanism design. Despite the natural setting, to our knowledge this is the first work to address it.

As we discuss in the paper, there is more than one notion of fairness that one might consider. Our primary notion of fairness—*k-fairness*—places a cap on the disutility of each agent and thus on the discrepancy between the disutilities of different agents (our secondary notion of fairness—competence penalty—plays only a secondary role). This notion of fairness is influenced by computer science (specifically, the max–min criterion of bandwidth allocation), but should not appear foreign to economists as well. For example, Hammond [4] shows that Rawls' maximin rule can be used to satisfy Arrow's conditions for a social welfare function when they are modified to incorporate comparisons among the agents. Our paper considers a related but different problem, because the selected social choice function must be implemented in a setting where agent types are privately known. As was shown in this paper, the self-interest of the agents limits the space of social choice functions that we can implement.

Of course, other notions could be studied, and this is an avenue for future research. Alternatively, we could move the goal of fairness from the set of goals of the mechanism designer to the utility functions of the agents. Although this models somewhat different situations than those that we have in mind and that were illustrated in the introduction, it can be interesting as well. Each agent's utility function could then be a function of not only the outcome for this agent and its own type, but also of the entire vector of declared types and the outcomes for all agents. A body of research already exists that considers fairness in this manner, where it is typically examined in the context of specific games such as the ultimatum or dictator game. For example, Ref. [2] presents a model in which the utility of each agent depends on equity, reciprocity (for past cooperation or the failure to do so), and the agent's relative position among all agents. It would be interesting to approach this setting from a mechanism-design perspective.

Returning to the alternative interpretation of our setting as an auction setting, our results concern the fair division of the payment made by the winning bidder. In order to achieve efficiency the winning bidder is charged the second-highest bid. As is well known, this amount cannot then simply be redistributed amongst the agents, because the agent with the second-highest bid would then have incentive to not bid truthfully (specifically, to bid higher). Instead, the optimal redistribution rule is to pay each agent the second-highest bid *of the other agents*, achieving 3-fairness. The meaning of the competence penalty in this setting is that the agent with the second-highest bid receives a smaller payment than that of the agents who value the good less.

We concentrated on the single task setting, while also showing how the results can be generalized to multiple tasks. As we move to a number of tasks much greater than two, the computational cost of determining allocation and payment rules can become prohibitive. Similar problems are faced in winner determination of combinatorial auctions, which have attracted much interest in recent years; solutions range from more efficient algorithms [5,9] to a shift to iterative mechanisms [7]. Exploring these directions in conjunction with the fairness considerations of this paper presents another opportunity for future work.

Appendix

Proof of Lemma 1. Proof by construction. We describe a transformation which takes any mechanism Γ (defined by f and its constitutive g and t) that satisfies IC, k-fairness and ND, and returns a mechanism Γ' (defined by f', g', and t'), which satisfies IC, k-fairness, ND, and EE.

We initialize Γ' to be identical to Γ, and then make any necessary changes. Because Γ satisfies IC, we know that $\hat{v} = v$. For each v, apply the following transformation. If $g_1(v) = 1$, then Γ' already satisfies EE and we are done for this particular v. Otherwise, we need to alter g' so that the task is instead assigned to agent 1, and alter t' to compensate for this reassignment. Call the service provider in Γ agent j (that is, $g_j(v) = 1$). To secure EE, we set $g'_1(v) = 1$ and $g'_j(v) = 0$. Then, set $t'_1(v) = t_1(v) - v_1$ and $t'_j(v) = t_j(v) + v_j$. The changes for g' and t' imply that for all i and v, the utility for agent i is equal in both mechanisms. That is, $u_i(f(v), v_i) = u_i(f'(v), v_i)$. We can then prove IC by contradiction. Assume that Γ' is not IC; then, there must exist some i, v, and v'_i such that $u_i(f'(v'_i, v_{-i}), v_i) > u_i(f'(v_i, v_{-i}), v_i)$. Because our transformation does not alter utilities, this inequality implies that $u_i(f(v'_i, v_{-i}), v_i) > u_i(f(v_i, v_{-i}), v_i)$, contradicting the assumption that Γ satisfies IC. The fact that utility remains constant implies that the k-fairness property of Γ is preserved in Γ'. Finally, ND continues to hold because $\sum_i t'_i(v) = \sum_i t_i(v) - v_1 + v_j \geqslant \sum_i t_i(v)$. $\square$

Proof of Theorem 1. We will use a proof by contradiction that works for all $n \geqslant 2$. Assume that a mechanism does exist for a given $n \geqslant 2$ that satisfies IC, 2-fairness, and ND. Consider a vector v in which $v_1 < v_2$. We will show a lower bound on the payment to the service provider, and then an upper bound on the payments made by the other agents. These bounds will guarantee that ND cannot be satisfied.

Because of IC, we know that all agents other than agent 1 declare truthfully (that is, $\hat{v}_{-1} = v_{-1}$). Lemma 1 tells us that if a satisfying mechanism does exist, then there must exist one that satisfies EE. Thus, we can assume that if agent 1 declares truthfully, then it will be assigned the task (that is, $g_1(v_1, v_{-1}) = 1$). We now show that regardless of the particular value of v_1, subject to the constraint that $v_1 < v_2$, agent 1's payment, $t_1(v_1, v_{-1})$, must be constant. Otherwise, there must be two types $v_1', v_1'' < v_2$ such that $t_1(v_1', v_{-1}) > t_1(v_1'', v_{-1})$. Because agent 1's utility function is $u_1(f(\hat{v}_1, v_{-1}), v_1) = -v_1 - t_1(\hat{v}_1, v_{-1})$, this agent would have incentive to falsely declare v_1'' when its true type is v_1', violating IC.

Furthermore, we can show an upper bound of $t_1(v_1, v_{-1}) \leqslant -\frac{n-1}{n} v_2$ on this constant value. If this bound did not hold, then $t_1(v_1, v_{-1}) = -\frac{n-1}{n} v_2 + \varepsilon$, for some $\varepsilon > 0$. Consider the possibility of $v_1 = v_2 - \delta$, where $\delta < \varepsilon$. Then, $u_1(f(v), v_1) = -(v_2 - \delta) - (-\frac{n-1}{n} v_2 + \varepsilon) = -v_2/n + \delta - \varepsilon < -v_2/n$, violating 2-fairness.

Next, consider any other agent i, where $i \neq 1$. Holding the declarations of the other agents fixed at $\hat{v}_{-i} = v_{-i}$, it must be the case that $t_i(v_i, v_{-i})$ is constant for all possible types of agent i such that $v_i > v_1$. That is, as long as agent i's type would not make it the service in the case of a truthful declaration, it must always pay the same amount. If this were not true, then there must exist two types $v_i', v_i'' > v_1$ such that $t_i(v_i', v_{-i}) > t_i(v_i'', v_{-i})$. Since $u_i(f(\hat{v}_i, v_{-i}), v_i) = -t_i(\hat{v}_i, v_{-i})$, agent i would have incentive to falsely declare v_i'' when its true type is v_i', violating IC. Next, we can show an upper bound on this constant payment for all $v_i > v_1$: $t_i(v_i, v_{-i}) \leqslant v_1/n$. If this were not true, then $t_i(v_i, v_{-i}) = (v_1 + \varepsilon)/n$ for some $\varepsilon > 0$. Consider the possibility of $v_2 = v_1 + \delta$, where $\delta < \varepsilon$. In this case, $u_i(f(v), v_1) = -(v_1 + \varepsilon)/n < -v_2/n$, violating 2-fairness.

Since the above argument holds for the $n-1$ agents other than the service provider (agent 1) when all agents declare their type truthfully, we have an upper bound on the net payments to the center: $\sum_{i \neq 1} t_i(v) + t_1(v) \leqslant (n-1) \cdot \frac{v_1}{n} - \frac{n-1}{n} v_2$. Since $v_2 > v_1$, ND is not satisfied, reaching a contradiction. $\square$

Proof of Theorem 2. We start by proving IC. Consider any possible declaration vector $\hat{v}$. We need to show that any agent i whose declaration is truthful ($\hat{v}_i = v_i$) could not increase its utility by making an alternate declaration, holding $\hat{v}_{-i}$ constant. Since we will need to talk about properties of the original declaration vector $\hat{v}$ throughout the proof, we will use $\hat{v}_i'$ to denote this alternate declaration.

There are three possible "classes" that a truthful agent i could fall into:

(1) $v_i = \hat{v}_{[1]}$ and $g_i(\hat{v}) = 1$,
(2) $v_i = \hat{v}_{[2]}$ and $g_i(\hat{v}) = 0$,
(3) $v_i > \hat{v}_{[2]}$ and $g_i(\hat{v}) = 0$.

In the first class, $u_i(f(v_i, \hat{v}_{-i}), v_i) = -v_i + \hat{v}_{[2]} - \frac{\hat{v}_{[3]}}{n}$. We know that $u_i(f(v_i, \hat{v}_{-i}), v_i) \geqslant -\frac{\hat{v}_{[3]}}{n}$, since $g_i(v_i, \hat{v}_{-i}) = 1$ implies that $v_i \leqslant \hat{v}_{[2]}$. Any alternate declaration such that $\hat{v}_i' < \hat{v}_{[2]}$ would not change $g(\cdot)$ or $t(\cdot)$, and thus would not change agent i's utility. For all $\hat{v}_i' > \hat{v}_{[2]}$, agent i's utility is $u_i(f(\hat{v}_i', \hat{v}_{-i}), v_i) = -\frac{\hat{v}_{[3]}}{n}$, because it is never chosen as the service provider. The final possibility of $\hat{v}_i' = \hat{v}_{[2]}$ is covered by one of the previous two cases, depending on how the tie is broken. Thus, there does not exist a $\hat{v}_i'$ such that $u_i(f(\hat{v}_i', \hat{v}_{-i}), v_i) > u_i(f(v_i, \hat{v}_{-i}), v_i)$.

In the second class, $u_i(f(v_i, \hat{v}_{-i}), v_i) = -\frac{\hat{v}_{[3]}}{n}$. If $\hat{v}_i{}' < \hat{v}_1$, then $u_i(f(\hat{v}_i{}', \hat{v}_{-i}), v_i) = -v_i + \hat{v}_{[1]} - \frac{\hat{v}_{[3]}}{n}$, because agent i becomes the service provider and $\hat{v}_{[1]}$ becomes the second highest declaration. In this case we know that $u_i(f(\hat{v}_i{}', \hat{v}_{-i}), v_i) < -\frac{\hat{v}_{[3]}}{n}$, since $g_i(v_i, \hat{v}_{-i}) = 0$ implies that $v_i > \hat{v}_{[1]}$. Alternatively, if $\hat{v}_i{}' > \hat{v}_{[1]}$, then $u_i(f(\hat{v}_i{}', \hat{v}_{-i}), v_i) = -\frac{\hat{v}_{[3]}}{n}$. The case of $\hat{v}_i{}' = \hat{v}_{[1]}$ is covered by one of the previous two cases, depending on how the tie is broken. Thus, for all $\hat{v}_i{}'$: $u_i(f(\hat{v}_i{}', \hat{v}_{-i}), v_i) \leqslant u_i(f(v_i, \hat{v}_{-i}), v_i)$.

IC in the third class is shown exactly the same way as it was in the second class. The only difference is that $\frac{\hat{v}_{[3]}}{n}$ is replaced by $\frac{\hat{v}_{[2]}}{n}$ in $u_i(\cdot)$ to reflect the change in the second lowest declaration of the other agents.

We now show that the remaining requirements are also satisfied. EE follows directly from IC and the definition of Fair3. Substituting v for $\hat{v}$ in equations we derived during the proof of IC, we have $u_i(f(v), v_i) \geqslant -\frac{v_{[3]}}{n}$ for an agent in class 1, $u_i(f(v), v_i) = -\frac{v_{[3]}}{n}$ for an agent in class 2, and $u_i(f(v), v_i) = -\frac{v_{[2]}}{n}$ for an agent in class 3, proving 3-fairness. Finally, for ND:

$$\begin{aligned}\sum_i t_i(v) &= t_1(v) + t_2(v) + \sum_{i \geqslant 3} t_i(v)\\ &= \left(-v_{[2]} + \frac{v_{[3]}}{n}\right) + \frac{v_{[3]}}{n} + \sum_{i \geqslant 3} \frac{v_{[2]}}{n}\\ &\geqslant -v_{[2]} + \frac{n}{n} v_{[2]}\\ &\geqslant 0. \qquad \square\end{aligned}$$

Proof of Theorem 3. The proof of this result for $n = 2$ follows directly from Theorem 1. We will prove by contradiction that it holds for all $n \geqslant 3$ using an argument that does not depend on the specific n chosen. Assume that a mechanism does exist for a given $n \geqslant 3$ that satisfies IC, n-fairness, and ND, and does not enforce a competence penalty. Consider the possibility that $v_1 < v_2$ holds in the vector v of true types, and that all agents declare truthfully. By Lemma 1, we can assume that $g_1(v) = 1$. (Recall that we have assumed an ordering on the agents from ease of exposition.)

The first step is to prove by induction that for each agent i such that $1 < i \leqslant n$ (i.e., the non-service providers), $t_i(v) \leqslant v_{i-1}/n$ must hold. Starting with the base case of $i = n$, we must show that $t_n(v) \leqslant v_{n-1}/n$. Because of n-fairness, $t_n(v) \leqslant v_n/n$, since $u_n(f(v), v_n) = -t_n(v)$. IC requires the mechanism to keep $t_n(\hat{v}_n, v_{-n})$ constant for all declarations of agent n subject to the constraint that $\hat{v}_n > v_{n-1}$. Otherwise, there must exist two types $v_n', v_n'' > v_{n-1}$ such that $t_n(v_n', v_{-n}) > t_n(v_n'', v_{-n})$. IC would then be violated because agent n has incentive to falsely declare v_n'' when its true type is v_n'. Next, we can show the desired upper bound on this constant payment for all $\hat{v}_n > v_{n-1}$: $t_n(\hat{v}_n, v_{-n}) \leqslant v_{n-1}/n$. If this were not true, then $t_n(\hat{v}_n, v_{-n}) = (v_{n-1} + \varepsilon)/n$ for some $\varepsilon > 0$. Consider the possibility of $v_n = v_{n-1} + \delta$, where $\delta < \varepsilon$. In this case, $t_n(v) > v_n/n$, contradicting a bound we derived above. Therefore, the base case holds.

We now prove the inductive step for each i in the range $1 < i < n$. By the inductive assumption, $t_{i+1}(v) \leqslant v_i/n$. We must show that $t_i(v) \leqslant v_{i-1}/n$. To avoid a *competence penalty*, $t_i(v) \leqslant t_{i+1}(v) \leqslant v_i/n$. Furthermore, $t_i(\hat{v}_i, v_{-i})$ must be constant for all $\hat{v}_i > v_{i-1}$ by the same incentive compatibility argument used above, because such a declaration cannot change the service provider for any $i > 1$. We can then show an upper bound of $t_i(\hat{v}_i, v_{-i}) \leqslant v_{i-1}/n$, also by a similar argument as above. Thus, the inductive step holds, and we can conclude that for each agent i such that $1 < i \leqslant n$, $t_i(v) \leqslant v_{i-1}/n$.

The second step is to show that these bounds prevent the mechanism from satisfying n-fairness. To do this, we use a possible instance of v that can be applied to any value of $n \geqslant 3$. Consider the case in which $v_1 = 0$, and for all $i > 1$, $v_i = n^2 + i$. Also, let v be the declared types of the agents. Since this v satisfies the constraint of $v_1 < v_2$, the bounds we just derived hold. Thus, the net payment from the non-service providers to the center is as follows:

$$
\begin{aligned}
\sum_{i=2}^{n} t_i(v) &\leqslant \sum_{i=2}^{n} v_{i-1}/n \\
&\leqslant 0 + \sum_{i=2}^{n-1} (n^2 + i)/n \\
&< \frac{n^2(n-2) + (n-1)(n)/2}{n} \\
&= n^2 - 1.5n - 0.5.
\end{aligned}
$$

Because of ND, we also have a bound on the payment from the center to the service provider: $-t_1(v) \leqslant \sum_{i=2}^{n} t_i(v) < n^2 - 1.5n - 0.5$. We also know that $-t_1(\hat{v}_1, v_{-1})$ must be constant for all $\hat{v}_1 < v_2$, by the same incentive compatibility used above. Thus, it must be the case that $-t_1(v'_1, v_{-1}) = -t_1(v) < n^2 - 1.5n - 0.5$, when the declared type of agent 1 is: $v'_1 = n^2 + 1$. However, it is possible that v'_1 is also the *true* type of agent 1. In this case, agent 1's utility is too large to satisfy n-fairness.

$$
\begin{aligned}
u_1(f(v'_1, v_{-1}), v'_1) &= -\, t_1(v'_1, v_{-1}) - v'_1 \\
&= -\, t_1(v_1, v_{-1}) - v'_1 \\
&< n^2 - 1.5n - 0.5 - (n^2 + 1) \\
&= -\, 1.5n - 1.5 \\
&= -\, \frac{1.5n^2 + 1.5n}{n} \\
&< -\, \frac{n^2 + n}{n} \\
&= -\, \frac{v_n}{n}.
\end{aligned}
$$

Since n-fairness is violated, we have reached a contradiction, and the proof is complete. $\square$

Proof of Theorem 4. The proof of IC is almost identical to that of mechanism Fair3. The only difference between the two mechanisms is that the "offset" payment $\left(\frac{\hat{v}_{-i_{[1]}}}{n}\right)$ is changed to use the lowest declaration among all other agents instead of the second lowest declaration from this set. Since the agent's own type still does not affect this term, IC follows in an identical fashion.

EE then follows directly from IC and the definition of BoundedFair1. For 1-fairness, we examine the two classes that an agent can fall into: the service provider, or a non-service provider. For the service provider, $u_i(f(v), v_i) = -v_i + v_{[2]} - \frac{v_{[2]}}{n}$ (because $\hat{v} = v$), which can be re-written as: $u_i(f(v), v_i) = -\frac{v_i}{n} - \frac{n-1}{n} v_i + \frac{n-1}{n} v_{[2]}$. Because this agent is the service provider and EE is satisfied, we know that $v_i = v_{[1]} \leqslant v_{[2]}$ and thus that $u_i(f(v), v_i) \geqslant -\frac{v_{[1]}}{n}$. For a non-service provider, $u_i(f(v), v_i) = -\frac{v_{[1]}}{n}$. These two facts, combined with IC, imply 1-fairness.

Finally, to show $(v_{[2]} - v_{[1]})$-BD, we carry out the following calculations.

$$\begin{aligned}
\sum_i t_i(v) &= t_1(v) + \sum_{i \neq 1} t_i(v) \\
&= -v_{[2]} + \frac{v_{[2]}}{n} + \sum_{i \neq 1} \frac{v_{[1]}}{n} \\
&= -\frac{n-1}{n} v_{[2]} + \frac{n-1}{n} v_{[1]} \\
&\geqslant -(v_{[2]} - v_{[1]}). \qquad \square
\end{aligned}$$

References

[1] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[2] G. Bolton, A. Ockenfels, Erc: a theory of equity, reciprocity and competition, Amer. Econ. Rev. 90 (2000) 166–193.
[3] T. Groves, Incentives in teams, Econometrica 41 (1973) 617–631.
[4] P. Hammond, Equity, arrow's conditions, and rawls' difference principle, Econometrica 44 (1976) 793–804.
[5] K. Leyton-Brown, Y. Shoham, M. Tennenholtz, An algorithm for multi-unit combinatorial auctions, in: AAAI, Austin, TX, 2000, pp. 56–61.
[6] A. Mas-Colell, M. Whinston, J. Green, Microeconomic Theory, Oxford University Press, Oxford, 1995.
[7] D.C. Parkes, L.H. Ungar, Iterative combinatorial auctions: theory and practice, in: AAAI, Austin, TX, 2000, pp. 74–81.
[8] J. Rawls, A Theory of Justice, Belknap Press, Cambridge, MA, 1971.
[9] T. Sandholm, Algorithm for optimal winner determination in combinatorial auctions, Artif. Intell. 135 (1–2) (2002) 1–54.