DEPARTMENT OF FOOD AND RESOURCE ECONOMICS
UNIVERSITY OF COPENHAGEN

IFRO Working Paper

Decentralized Task Coordination

Jens Gudmundsson
Jens Leth Hougaard
Trine Tornøe Platz

2020 / 11

IFRO Working Paper 2020 / 11

Decentralized Task Coordination

Authors: Jens Gudmundsson, Jens Leth Hougaard, Trine Tornøe Platz

JEL-classification: C72, C78, D47, D63, D78

Published October 2020

This work is supported by the Center for Blockchains and Electronic Markets funded by the Carlsberg Foundation under grant no. CF18-1112.

See the full series IFRO Working Paper here:
www.ifro.ku.dk/english/publications/ifro_series/working_papers/

Department of Food and Resource Economics (IFRO)
University of Copenhagen
Rolighedsvej 25
DK 1958 Frederiksberg DENMARK
www.ifro.ku.dk/english/

Decentralized Task Coordination*

Jens Gudmundsson[1], Jens Leth Hougaard[1,2], and Trine Tornøe Platz[1]

[1]Department of Food and Resource Economics, University of Copenhagen, Denmark
[2]NYU-Shanghai, China

October 19, 2020

Abstract

We study decentralized task coordination. Tasks are of varying complexity and agents asymmetric: agents capable of completing high-level tasks may also take on tasks originally contracted by lower-level agents, facilitating system-wide cost reductions. We suggest a family of decentralized two-stage mechanisms in which agents first announce preferred individual workloads and then bargain over the induced joint cost savings. The second-stage negotiations depend on the first-stage announcements as specified through the mechanism's *recognition function*. We characterize mechanisms that incentivize cost-effective task allocation and further single out a particular mechanism, which additionally ensures a fair distribution of the system-wide cost savings.

Keywords Decentralized mechanisms · Implementation · Bargaining · Consistency · Blockchain

JEL Classification C72 · C78 · D47 · D63 · D78

1 Introduction

The ongoing development of the digital economy is enabling "smart" markets in which computational agents make decisions on behalf of organizations (e.g. Can, 2019), supported for instance by a blockchain-based infrastructure for increased transparency and decentralization of power (Abadi and Brunnermeier, 2018). It creates new avenues for cooperation between independent self-optimizing agents, with ample opportunities for mutual gains when supported by well-designed institutions and mechanisms. These mechanisms can be implemented by self-executing *smart contracts* (e.g. Gans, 2019; Catalini and Gans, 2020), automating transactions and eliminating the need for trusted third parties. The particular design of such mechanisms poses a serious challenge as one must account for issues pertaining to both incentives and fairness to ensure efficient and sustainable cooperation. However, when successful, it holds tremendous potential as automated agents can solve bottlenecks in almost all parts of the economy without distortion from third-party mediators, reducing both transaction and contract enforcement costs (see e.g. World Bank Group, 2020).

*This work is supported by the Center for Blockchains and Electronic Markets funded by the Carlsberg Foundation under grant no. CF18-1112. Emails: jg@ifro.ku.dk, jlh21@nyu.edu, and ttp@ifro.ku.dk.

Our focus is on creating and sharing systemic cost reductions from distributed task processing. Whereas most of the literature on job scheduling and task allocation concerns optimal solutions implemented in a centralized way by a "scheduler" or a trusted third party (see e.g. Brucker, 2004; Bertsimas and Farias, 2012; Hillier and Lieberman, 2021), we explore decentralized solutions. Specifically, we study cost-effective coordination among autonomous service providers ("agents", henceforth) who individually have contracted on and guaranteed the completion of certain tasks. Tasks and agents are heterogeneous: tasks differ for instance in service requirements while agents differ in the quality of service that they provide. Task assignment is thus highly asymmetric with almost all agents eligible for the most basic tasks but only the most complex agents, the service providers with the highest quality of service, capable of performing the most complex tasks. Other factors, such as privacy and security, may further influence task assignment and is succinctly represented through task/agent-specific capacity constraints. Typically, the uncoordinated contracts provide a suboptimal task allocation with system-wide cost savings attainable through careful reallocation. However, as taking on extra tasks is costly, and no agent can force another to do so, agents will only take on higher workloads if appropriately compensated. Hence, agents must decide on both an allocation as well as side payments to support it. In this way, the initial contracts serve as the the starting point from which we seek beneficial task reallocation as well as the default to which we revert if agents cannot reach agreement.

Inspired by Gudmundsson et al. (2019), we suggest a family of decentralized two-stage mechanisms to find and implement cost-effective allocations. In the mechanism's first stage, agents sequentially announce desired workloads subject to feasibility constraints; in the second, they bargain over the resulting cost reductions and settle on the side payments needed to support the reallocation. The bargaining process follows a random-proposer protocol, requiring unanimous consent to finalize the agreement (Binmore et al., 1986; Binmore, 1987; Baron and Ferejohn, 1989). This allows us to connect the two stages: the probability distribution over who gets to propose a solution in the second-stage negotiations will depend on the first-stage announcements. Specifically, an agent is "recognized" as the proposer with a certain probability, making the mechanism's *recognition function* a key design parameter. This function can depend on initial contracts, announced workloads, and capacity constraints, and it is exogenous to the agents. From a practical perspective, a mechanism can be hard-coded in a smart contract, and its recognition function can be implemented through a provably fair algorithm (compare blockchain-based gambling, see e.g. Min et al., 2019). Our objective is to identify recognition functions that incentivize agents to announce cost-effective first-stage workloads all the while ensuring a "fair" division of the cost reductions through the second-stage negotiations. Throughout, the primitives of the model—the initial contracts, capacities, and cost functions—are common knowledge.

Our mechanism induces a two-stage game, which we analyze through backward induction. In Proposition 1, we detail the agents' expected costs in all (stationary) subgame-perfect equilibria of the bargaining stage. Generally, there is a unique such equilibrium; in it, agents share the system-wide cost reductions in proportion to their recognition probabilities. This reveals a potential trade-off: it may be beneficial for an agent to announce an inefficient first-stage workload if it increases the agent's second-stage recognition probability (compare Holmstrom, 1982, Theorem 1); cost reductions are then smaller, but the agent acquires a larger share of them. Still, some recognition functions succeed in aligning individual incentives with cost-effective task allocation, encouraging agents to always announce efficient workloads. In this way, the hierarchical order of the agents from basic to complex together with the sequential decision making is used to overturn Holmstrom's (1982) impossibility. These recognition functions are said to satisfy *cost-effective implementation*. Proposition 2 gives a precise description of what *cost-effective implementation* implies for the recognition function: specifically, an agent's recognition probability has to be independent of the workloads announced by the agent herself as well as by the more complex agents.

While *cost-effective implementation* ensures desirable first-stage announcements, we turn to the literature on fair allocation to pin down desirable cost-savings distributions in the second-stage negotiations. We take an axiomatic approach and first turn to the notion of consistency (see e.g. Thomson, 2011, 2016). Loosely speaking, consistency extends the intuitive principles, which suggest the solution to a part of a problem, to the problem in its entirety. To explain the novelty of our approach, it is useful to separate the axiom in two parts: for a given problem, one typically specifies (i) a reduced problem and (ii) a relation between the original and the reduced problem. In our case, *consistency* pertains to the situation in which the probability for agent 1—the most basic service provider—has been settled; the agent's excess of contracted, but not yet completed, tasks are then distributed to the others to form the reduced problem. However, without compelling reasons to distribute the excess in one way rather than another, we do not specify a reduced problem for (i). Yet, there is a clear-cut relation that maintains the principle of (ii), namely that the remaining agents' relative recognition probabilities should be unchanged in the reduced problem. In this way, our *consistency* axiom asserts that there should exist *some* reduced problem with respect to which the recognition probabilities are unchanged when adjusting the probability mass accordingly. Figure 1 illustrates this approach.[1]

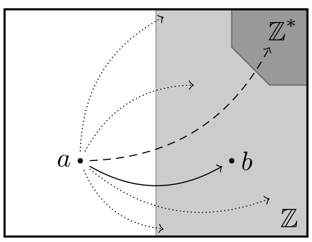

Figure 1: The original problem a and its possible reduced problems $\mathbb{Z}$ (light). The typical approach (solid) relates a to a specified problem b. Our *consistency* (dotted) only asserts that the relation should be maintained with respect to some reduced problem. However, to be compatible with *cost-effective implementation* and *top-agent proportionality*, Proposition 4 shows (dashed) that *consistent* recognition functions have to relate to problems in $\mathbb{Z}^*$ (dark).

A consequence of *cost-effective implementation* through Proposition 2 is that the recognition probability for agent 1 is independent of the first-stage announcements. We then turn to another well-documented principle in fair allocation, namely proportionality: *top-agent proportionality* asserts that a fair recognition probability for agent 1 equals her share of the initial contracts. Proposition 3 then identifies the functional form of recognition functions that satisfy *consistency* and *top-agent proportionality.* Thereafter, Proposition 4 pertains to the intersection of the two prior results: it details the reduced problems for which a function can be consistent so it remains compatible with *cost-effective implementation* and *top-agent proportionality.* Finally, building on Proposition 4, Theorem 1 provides an axiomatic characterization of a particular recognition function p^*. This first settles agent 1's probability to 1's share of the contracted tasks; it then continues recursively (order $2, 3, \dots$), awarding agent i a share of the remaining probability mass that equals i's maximal announcement in proportion to the total remaining tasks given the prior agents' announcements. Theorem 1 shows that p^* is the unique recognition function to satisfy *cost-effective implementation*, *consistency*, and *top-agent proportionality.* Thus, it succeeds both in incentivizing agents to announce cost-effective first-stage workloads as well as in distributing the cost reductions fairly through the second-stage negotiations.

Our contribution relates to several strands of literature. The operations research and computer science literature abound with job-scheduling problems of many different kinds (see e.g. Brucker, 2004). In a clas-

[1] Our approach is related to parameterizing axioms (see Thomson, 2019), say through "consistency parameterized by the set of reduced problems X". This would require the original problem to have a specified relation to some member of X. Our *consistency* axiom sets X to the entire set of reduced problems, the conventional notion sets X to a single problem, and Proposition 4 concerns the case $X = \mathbb{Z}^*$ as defined in the main text.

sic job-scheduling problem, agents control machines that have to perform a number of jobs with different lengths as well as starting and ending times. Given machine capacities, the scheduler's problem typically consists of minimizing the makespan (i.e., minimizing the resource cost). In particular, a recent and very popular topic is scheduling of cloud computing, with numerous papers analyzing various forms of resource allocation mechanisms (see e.g. Azar et al., 2015; Li et al., 2016; Bao et al., 2018). For instance, Huang et al. (2015) consider jobs submitted to cloud computing clusters and Chen et al. (2019) consider jobs across geo-distributed datacenters, both with a focus on optimal centralized resource allocation and fairness when utility is sensitive to completion times. Zhang et al. (2015) and Zhou et al. (2017) analyze the same problem using market mechanisms in the form of different auction designs. A branch of the scheduling literature considers the benefits of distributed processing, typically dubbed "task allocation problems" (see e.g. Dutta et al., 1982; Ernst et al., 2006): a set of tasks is centrally assigned to a set of (capacitated) processors such that total processing cost is minimized. Due to problem complexity in real-world applications, focus is primarily on optimization and heuristic algorithms. Lastly, Tawarmalani et al. (2009) consider allocating objects in a network of caches. Nodes in the network face external object demands, analogous to our agents' initial contracts, which they can meet by either offering the objects themselves or by relying on neighboring nodes to do so. As nodes have capacity constraints, they have to think carefully about which objects to hold. Tawarmalani et al. (2009) design an auction mechanism that settles side payments such that a cost-effective allocation is reached.

The economic literature has mainly been concerned with fairness and incentives in queueing (e.g. Chun, 2016) and scheduling (e.g. Moulin, 2007). In short, a machine, which serves one job at a time, is shared by users with jobs of arbitrary length and waiting-time costs, and the question is how users share the joint externality by suitable side payments. Recently, Bahel and Trudeau (2019) consider a version of classic job scheduling with a focus on fair division of the efficient cost using the framework of cooperative game theory. They provide a characterization of stable cost allocations in the sense of the core as well as axiomatic characterizations of two allocation rules. We, on the other hand, maintain a similar focus on fair division of welfare, but analyze how allocation of welfare influences agents' incentives via decentralized, non-cooperative mechanisms. Hence, there is no centralized scheduling done by a planner and agents are autonomous entities, organizing their own production schedule through the mechanism. Lastly, our model provides a generalization in several directions of the river-sharing model studied by Gudmundsson et al. (2019), who reallocate a homogeneous resource among an ordered set of agents. In contrast, our tasks are heterogeneous, agents may perform some, but perhaps not all, task types, and reallocation is constrained by the agents' capacities. In the river context, welfare gains require collaboration between high-inflow and high-consumption countries and is facilitated by intermediate transfer countries; in the present context, there is no analogue to an "intermediate transfer country": while water is bound to flow downstream, tasks do not obey such a physical restriction. This is also the reason that we opt for the weak form of *consistency*. Thus, while there is an overlap in our axiomatic approaches, the conditions imposed here are novel and lead us to new, interesting results.

The paper is organized as follows. Section 2 defines the formal model and the mechanisms. Section 3 introduces the axiomatic analysis. Our results are contained in Section 4. Section 5 closes with final remarks on potential modifications of the model. Technical remarks and proofs are postponed to the Appendix.

2 Model

We study the interaction of a group of heterogeneous agents, broadly interpreted as service providers, set to complete a number of tasks of varying complexity. We take as given that the agents have incentives to complete all tasks and that they will do so as, say, the negative long-term effects of a contract breach exceeds the one-time cost of performing the tasks. While the primitives of the model are common knowledge, the agents are independent and make independent decisions, each seeking to minimize their own cost. In this way, there is a joint interest in concerting a cost-effective task allocation, but it is only attainable if all agents obtain a fair and agreeable share of the system-wide cost reductions it entails. We leave open the relation between the agents: the collaboration may be between firms, such as airline alliances (Hu et al., 2013), or cross-functional within a firm (Kouvelis and Lariviere, 2000) in which senior staff may be overqualified yet able to perform tasks intended for junior staff.

2.1 Preliminaries

There is a set of **agents** $N = \{1, \dots, n\}$ and a set of **task types** $T = \{1, \dots, m\}$, both ordered by complexity: agent 1 and type 1 are the most basic, whereas n and m are the most complex. We assume throughout that tasks are perfectly divisible. Each agent i has signed a contract on the completion of $d_{it} \geq 0$ tasks of type t; we let $\mathbf{D} \in \mathbb{R}^{n\times m}_{\geq 0}$ denote the matrix specifying the contracted tasks. This will serve also as the "default allocation" and the disagreement point in the forthcoming negotiations: if agreement to reallocate tasks differently is not reached, then we revert to $\mathbf{D}$. Let $T_i = \{t \in T : d_{it} > 0\}$ denote agent i's contracted task types. Easier tasks may in some cases be performed at a satisfactory level also when delegated to more complex agents. This is captured by the **capacity matrix** $\mathbf{C} \in \mathbb{R}^{n\times m}_{\geq 0}$ such that $\mathbf{C} \geq \mathbf{D}$,[2] specifying that agent i can perform at most $c_{it} \geq 0$ tasks of type t. If i is ineligible to perform tasks of type t, then $c_{it} = 0$. The purpose of redistributing tasks is to lower systemic costs. Specifically, for each agent i, the strictly convex and increasing **cost function** $f_i : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ specifies the cost $f_i(z)$ to agent i of completing z tasks (regardless of types). In summary, the primitives of the model are $\mathbf{D} \in \mathbb{R}^{n\times m}_{\geq 0}$, $\mathbf{C} \in \mathbb{R}^{n\times m}_{\geq 0}$, and $f = (f_1, \dots, f_n)$.

An **allocation** $\mathbf{X} \in \mathbb{R}^{n\times m}_{\geq 0}$ specifies that agent i completes $x_{it} \geq 0$ tasks of type t. Moreover, tasks are done only by eligible agents and all tasks are completed, so the set of allocations is $\mathcal{X}(\mathbf{D}, \mathbf{C}) = \{\mathbf{X} \in \mathbb{R}^{n\times m}_{\geq 0} : \mathbf{X} \leq \mathbf{C}$ and $x_1 + \dots + x_n = d_1 + \dots + d_n\}$. The **total cost** for allocation $\mathbf{X}$ is $F(\mathbf{X}) = \sum_i f_i(\sum_t x_{it})$. An allocation that minimizes F is **cost effective**. Given $\mathbf{D}$, $\mathbf{C}$, and f, let $\mathcal{E}(\mathbf{D}, \mathbf{C}, f) \subseteq \mathcal{X}(\mathbf{D}, \mathbf{C})$ denote the set of cost-effective allocations.[3] The **cost savings** from redistributing tasks from $\mathbf{D}$ to $\mathbf{X}$ are $\Delta(\mathbf{X}, \mathbf{D}, f) = F(\mathbf{D}) - F(\mathbf{X})$. Given that $\mathbf{X}$ may require some agents to undertake more tasks than contracted on, they need to be compensated for doing so. This is achieved through side payments $\pi \in \mathbb{R}^n$ such that $\pi_1 + \dots + \pi_n = 0$. The (net) cost to agent i of allocation $\mathbf{X}$ and payments π is $f_i(\sum_t x_{it}) - \pi_i$. As individual side payments are unbounded but jointly add to zero, the set of attainable vectors $U(\mathbf{X}, \mathbf{D}, \mathbf{C})$, or $U(\mathbf{X})$ for short, contains all ways of sharing the total cost $F(\mathbf{X})$: $U(\mathbf{X}) = \{u \in \mathbb{R}^n : \sum_i u_i = F(\mathbf{X})\}$.

A special subdomain of problems is interesting from both a theoretical and practical perspective. Generically, no two agents/service providers are identical in practice in the sense that they all provide service of differing quality. This puts additional restrictions on the problem, namely that no two agents contract on the same tasks, $T_i \cap T_j = \emptyset$, and, for agents $i < j$ and task type $t \in T_j$, $c_{it} = 0$. To distinguish this special

[2] We use element-wise comparisons. Hence, $\mathbf{C} \geq \mathbf{D}$ means $c_i \geq d_i$ for each i; this in turn means $c_{it} \geq d_{it}$ for each t.

[3] When there are multiple cost-effective allocations, say $\mathbf{X}$ and $\mathbf{Y}$, then, for each agent i, $\sum_t x_{it} = \sum_t y_{it}$.

case from the general one, we let $\mathbb{D}$ denote the "extended" domain defined originally and $\tilde{\mathbb{D}} \subset \mathbb{D}$ denote the special case just referred to. That is, $\tilde{\mathbb{D}} = \{\mathbf{D} \in \mathbb{D} : i \neq j \implies T_i \cap T_j = \emptyset\}$. Next, we illustrate the model through a numerical example.

Example 1. Consider the five-agent, five-task example with $\mathbf{D}$ and $\mathbf{C}$ as follows:

$$\mathbf{D} = \begin{bmatrix} 9 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{C} = \begin{bmatrix} 9 & 0 & 0 & 0 & 0 \\ 2 & 5 & 0 & 0 & 0 \\ 2 & 2 & 4 & 0 & 0 \\ 1 & 2 & 1 & 2 & 0 \\ 1 & 1 & 2 & 1 & 1 \end{bmatrix}.$$

Each agent has contracted on a single task type and $\mathbf{D} \in \tilde{\mathbb{D}}$. The agents can complete their obligations as $\mathbf{C} \geq \mathbf{D}$ and some tasks, but not all, can be delegated from lower- to higher-level agents. Assuming identical cost functions $f_1 = \cdots = f_5$, two among many cost-effective allocations are $\mathbf{X}$ and $\mathbf{Y}$ as follows:

$$\mathbf{X} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 1 & 1 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix} \text{ and } \mathbf{Y} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 \\ 0 & 1 & 2 & 0 & 1 \end{bmatrix}.$$

Thus, agent 1 completes five tasks while the other agents complete four each. As an example, with $f_i(z) = z^2$, the cost savings amount to $\Delta(\mathbf{X}) = F(\mathbf{D}) - F(\mathbf{X}) = (81 + 25 + 16 + 4 + 1) - (25 + 16 + 16 + 16 + 16) = 38$. $\circ$

2.2 Two-stage mechanisms

Our aim is to identify decentralized methods which ensure that agents are incentivized to cost-effectively redistribute tasks without coordinating through a trusted third party. For this, we look to the mechanisms introduced in Gudmundsson et al. (2019). These operate in two stages. First, agents declare their desired workloads. As it may be possible to delegate tasks from lower- to higher-level agents, agents make their announcements sequentially: agent 1 chooses first; agent 2 chooses second, and the tasks available to choose from will depend on 1's choice; and so on until agent n is left to take on the remaining tasks.[4] Still, this is merely a declaration of interest: in the second stage, every agent has a chance to veto the allocation and revert to $\mathbf{D}$. In this way, the first stage ends with some announced allocation $\mathbf{X}$ while the second pertains to bargaining over the total costs $F(\mathbf{X})$. In particular, one agent proposes a vector of costs $u \in U(\mathbf{X})$ for the others to accept or reject. Only on unanimous acceptance is the allocation actually realized: if the proposal is vetoed, then there is a new round of negotiations with a new proposal to evaluate.

A central characteristic of the mechanism is *who* gets to propose. The proposer holds an advantage—they can assign the highest cost shares acceptable to all other agents and thereby minimize their own share. Following the literature on fair allocation, we take as given that all agents should have a fair chance of

[4] For practical purposes, the process may be expedited if agents make simultaneous announcements. Over- and under-demanded tasks can then be assigned as a function of the announcements; compare for instance the "lexicographic allocation" in Cachon and Lariviere (1999, page 1097). While Cachon and Lariviere are unable to achieve cost effectiveness in equilibrium (Cachon and Lariviere, 1999, Theorem 4) in a model featuring multiple retailers announcing desired quantities to a supplier, our second-stage negotiations help to overturn this result.

obtaining this advantage: for instance, we may want to favor agents who have contracted on many tasks or who have announced that they are willing to take on many tasks. In this way, the mechanism chooses the proposer as a function of $\mathbf{X}$, $\mathbf{D}$, and $\mathbf{C}$; specifically, agent i is chosen with probability $p_i(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in [0, 1]$. We call p the *recognition function* and $p_i(\mathbf{X}, \mathbf{D}, \mathbf{C})$ agent i's *recognition probability.*

Within the context of computational tasks and services, we may expect algorithms to compute the announcements and handle the thereupon following bargaining stage on behalf of the service providers involved. Under such circumstances, we would expect agreement to be reached almost instantaneously. Next, we describe the two stages formally.

Stage 1: Sequential announcements of desired workloads. Agents sequentially choose task vectors $x_j = (x_{j1}, \ldots, x_{jm})$. Choices are limited by the agent's capacity c_j, task availability given prior choices x_i and the contracts d_i for $i < j$, and the remaining capacities c_k for $k > j$. The agent should be able to perform their announced tasks, the choice should be compatible with the prior choices, and the tasks that reamin should be managable by the higher-level agents. Let $\mathbf{X}$ denote the allocation eventually formed by the individual announcements $x_1, \ldots, x_n$. Formally,

1. Agent 1 chooses $x_1 \leq c_1$ such that $x_1 \leq d_1$ and $x_1 + \sum_{k>1} c_k \geq \sum_i d_i$.

$\vdots$

j. Agent j chooses $x_j \leq c_j$ such that $\sum_{i \leq j} x_i \leq \sum_{i \leq j} d_i$ and $\sum_{i \leq j} x_i + \sum_{k>j} c_k \geq \sum_i d_i$.

$\vdots$

n. Agent n chooses x_n such that $\sum_i x_i = \sum_i d_i$.

The latter inequalities provide a lower bound on each agent's announcement for the more complex agents to be able to complete the remaining tasks. There is also an upper bound: given the announcements $\mathbf{X}_{<j}$ by agents prior to agent j, the maximal announcement that j can make is[5]

$$\bar{x}_j = \min\{\textstyle\sum_{i \leq j} d_i - \sum_{i<j} x_i, c_j\}.$$

Agent j cannot go beyond c_j for capacity reasons nor beyond the total tasks available given the prior announcements.

Stage 2: Bargaining on side payments. Agents bargain, possibly indefinitely, on how to share the costs using a random-proposer protocol. If a proposal is rejected, then negotiations continue to the next round with probability δ and break down otherwise. We may view δ as a built-in feature of the mechanism used to speed up the negotiations: the lower δ, the more likely it is that negotiations break down and thus the riskier it is to reject a proposal. Formally, Stage 2 is as follows:

1. Some agent gets to propose a cost vector from $U(\mathbf{X})$. In particular, agent i is selected as proposer with probability $p_i(\mathbf{X}, \mathbf{D}, \mathbf{C})$.[6]
2. In random order,[7] agents choose whether to accept or reject the proposed costs.

[5] The minimum is taken element-wise: for each type t, $\bar{x}_{jt} = \min\{\sum_{i \leq j} d_{it} - \sum_{i<j} x_{it}, c_{jt}\}$.

[6] We may also view it as all agents choosing cost vectors simultaneously, but that the acceptance decision (steps 2 and onward) only concerns the vector announced by the agent chosen to be the proposer.

[7] By 'random order' we mean that the order is chosen randomly from all orders on the set of agents. Thus, for each agent, there is positive probability that they are the last to decide on whether to accept or reject the proposal.

3. If unanimously accepted, the proposal is implemented and the negotiations end.
4. Otherwise, if the proposal is rejected by at least one agent:
   (a) With probability δ, the negotiations proceed to the next round, returning to Step 1 above with a possibly different proposer drawn again using the distribution $p(\mathbf{X}, \mathbf{D}, \mathbf{C})$.
   (b) With probability $1-\delta$, the negotiations break down and end, reverting to the allocation $\mathbf{D}$ without side payments.

For tractability, we restrict attention to **stationary** strategies represented by a pair $(u^i, a_i) \in U(\mathbf{X}) \times \mathbb{R}$: agent i always proposes the cost vector u^i and always accepts a proposal in which their assigned cost does not exceed a_i.

3 Normative foundations for recognition functions

As the recognition function p ties the two stages of the mechanism together, it is a key design element that affects both agents' welfare and their incentives to distribute tasks in a cost-effective way. We now formally define recognition functions and introduce three axioms for such functions, inspired by the literature on fair allocation.

3.1 Recognition functions

Throughout, we fix the number of tasks m whereas the number of agents n may vary. An n-agent **problem** is described by a triple $(\mathbf{X}, \mathbf{D}, \mathbf{C})$ such that $\mathbf{X} \in \mathcal{X}(\mathbf{D}, \mathbf{C})$, $\mathbf{D} \in \mathbb{D}$, and $\mathbf{C} \geq \mathbf{D}$. Let $\mathcal{P}^n$ denote the set of n-agent problems and $\tilde{\mathcal{P}}^n = \{(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n : \mathbf{D} \in \tilde{\mathbb{D}}\}$ denote the restricted domain of problems connected to $\tilde{\mathbb{D}}$. A recognition function p selects, for each population size $n \in \mathbb{N}$ and each n-agent problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$, a point $p(\mathbf{X}, \mathbf{D}, \mathbf{C}) = (p_1(\mathbf{X}, \mathbf{D}, \mathbf{C}), \ldots, p_n(\mathbf{X}, \mathbf{D}, \mathbf{C}))$ in the n-simplex. That is, $p_i(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in [0, 1]$ is agent i's recognition probability, and the probabilities add to one, $\sum_i p_i(\mathbf{X}, \mathbf{D}, \mathbf{C}) = 1$. Each recognition probability p_j is continuous in its inputs x_{it}, d_{it}, and c_{it}, so small input variations only have small probability effects.[8]

3.2 Cost-effective implementation

Our first axiom pertains to cost effectiveness. Take as given contracted tasks $\mathbf{D} \in \tilde{\mathbb{D}}$, a capacity matrix $\mathbf{C}$, a list of cost functions $f = (f_1, \ldots, f_n)$, and the associated cost-effective allocations $\mathcal{E}(\mathbf{D}, \mathbf{C}, f)$. As noted, the recognition function affects the incentives for the first-stage announcements. For instance, if agent i's recognition probability is increasing in i's own announcement, we may expect i to announce a high, possibly inefficient, workload. On the other hand, if i's probability is independent of i's own announcement, then i's incentives are better aligned with the common objective of minimizing costs: it is in i's interest to choose x_i to maximize the cost savings $\Delta(\mathbf{X}, \mathbf{D}, f)$. In this way, some recognition functions encourage agents to announce cost-effective workloads. The axiom *cost-effective implementation* requires exactly that.

Axiom 1 (Cost-effective implementation)**.** For each population size $n \in \mathbb{N}$, matrix $\mathbf{D} \in \tilde{\mathbb{D}}$, capacity matrix $\mathbf{C} \geq \mathbf{D}$, and list of cost functions $f = (f_1, \ldots, f_n)$, every subgame-perfect equilibrium of the two-stage game induced by the recognition function p entails cost-effective announcements: that is, in each such equilibrium, the first-stage announcement $\mathbf{X}$ is in $\mathcal{E}(\mathbf{D}, \mathbf{C}, f)$.

[8] See Gudmundsson et al. (2019, Appendix A) for an extended discussion on continuity in recognition functions.

The approach we take is familiar from the literature on implementation theory (see e.g. Maskin and Sjöström, 2002; Corchón, 2009). We seek a game form (here, the two-stage mechanisms represented by the recognition function) that combines with the agents' types (here, their cost functions) to induce a game with socially desirable equilibria (here, cost-effective allocations). While *cost-effective implementation* may appear a strong condition, it is immediate that it is satisfied by some recognition functions. A trivial example is to equalize probabilities through $p_j(\mathbf{X}, \mathbf{D}, \mathbf{C}) = 1/n$ for each agent j. As each agent's probability is independent on her own announcement, the agent will want to minimize total costs. However, there are also recognition functions that satisfy *cost-effective implementation* yet depend on the announcements $\mathbf{X}$; we take a closer look at these in Section 4 and Proposition 2.

3.3 Consistency

Typically, a consistency axiom (see e.g. Thomson, 2011, 2016) specifies a particular reduced problem and a condition that connects the reduced problem to the original problem. In our context, the reduced problem corresponds to the situation in which the first agent's probability has been settled. The tasks that this agent announces that she will not perform are made available to remaining eligible agents. The condition then asserts that the recognition probabilities should be unchanged when adjusting the probability mass accordingly. When as much as possible of agent 1's surplus tasks are passed on to agent 2, subsequently agent 3, and so forth, we obtain *consistency with direct transfer* as defined below. The index "≥ 2" indicates all but agent 1: for instance, $\mathbf{X}_{\geq 2}$ is the sub-matrix of $\mathbf{X} \equiv \mathbf{X}_{\geq 1}$ comprised of the announcements by all agents but agent 1. The matrix $\mathbf{Z} \in \mathbb{R}^{n \times m}$ specifies how the excess—the tasks that agent 1 has contracted on but announced she does not intend to perform—is split among the agents. For *consistency with direct transfer*, $\mathbf{Z}$ is such that all excess, up to capacity, is pushed to the next agent.

Axiom 2 (Consistency with direct transfer)**.** For each population size $n \in \mathbb{N}$, problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$, and agent $j \geq 2$,

$$p_j(\mathbf{X}, \mathbf{D}, \mathbf{C}) = (1 - p_1(\mathbf{X}, \mathbf{D}, \mathbf{C})) \cdot p_j(\mathbf{X}_{\geq 2}, \mathbf{D}_{\geq 2} + \hat{\mathbf{Z}}_{\geq 2}, \mathbf{C}_{\geq 2}),$$

where $\hat{z}_1 = 0$ and, for each agent $j \geq 2$, $\hat{z}_j = \min\{e_1 - x_1 - (\hat{z}_1 + \cdots + \hat{z}_{j-1}), c_j - d_j\}$.

The values $\hat{z}_1, \ldots, \hat{z}_n$ are defined recursively. We first set $\hat{z}_1 = 0$ and then assign as much as possible of agent 1's excess to agent 2: $\hat{z}_2 = \min\{d_1 - x_1, c_2 - d_2\}$. We can view this as two cases: if $d_{1t} + d_{2t} \leq x_{1t} + c_{2t}$, then we fully specify $\hat{\mathbf{Z}}$ (for that type t) through $\hat{z}_{2t} = d_{1t} - x_{1t}$ and $\hat{z}_{3t} = \cdots = \hat{z}_{nt} = 0$; otherwise, we set $\hat{z}_{2t} = c_{2t} - d_{2t}$ and continue to the next agent to distribute the remaining excess.

In general, for each agent $j \geq 2$, we check whether $\sum_{i \leq j} d_{it} \leq \sum_{i<j} x_{it} + c_{jt}$. If so, building on the values $\hat{z}_{1t}, \ldots, \hat{z}_{j-1,t}$ determined in previous steps, we set $\sum_{i \leq j} \hat{z}_{it} = d_{1t} - x_{1t}$ and $\hat{z}_{j+1,t} = \cdots = \hat{z}_{nt} = 0$. Otherwise, we set $\hat{z}_{jt} = c_{jt} - d_{jt}$ and continue to the next agent. If we reach agent n, then necessarily $\sum_i d_{it} = \sum_i x_{it} \leq \sum_{i<n} x_{it} + c_{nt}$, so we fall into the first case, ensuring that $\hat{\mathbf{Z}}$ indeed is well-defined.

Consistency with direct transfer is a natural starting point but not completely uncontroversial. After all, tasks left over by agent 1 need not be passed on to the next agent capable of performing them; they may instead, for instance, be split evenly among all agents capable of performing them. For this reason, how the excess tasks are reassigned is not immediately clear. And indeed, how they are reassigned may affect the recognition probabilities: if all excess goes to agent 2, then that increases the relative importance of agent 2 and perhaps also her recognition probability; if instead only a small amount goes to agent 2, then, analogously, her recognition probability may decline. Therefore, we instead impose a very weak form of consistency. In what follows, *consistency* only asserts that there should exist some reduced problem with respect to which the recognition probabilities are unchanged when adjusting the probability mass accordingly. Formally, we let $\mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C})$ denote the set of distributions of agent 1's excess at $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$:

$$\mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C}) = \left\{ \mathbf{Z} \in \mathbb{R}^{n \times m}_{\geq 0} : z_1 = \vec{0} \text{ and } (\mathbf{X}_{\geq 2}, \mathbf{D}_{\geq 2} + \mathbf{Z}_{\geq 2}, \mathbf{C}_{\geq 2}) \in \mathcal{P}^{n-1} \right\}.$$

Axiom 2* (Consistency)**.** For each population size $n \in \mathbb{N}$, problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$, and agent $i = 2, \ldots, n$,

$$p_i(\mathbf{X}, \mathbf{D}, \mathbf{C}) = (1 - p_1(\mathbf{X}, \mathbf{D}, \mathbf{C})) \cdot p_i(\mathbf{X}_{\geq 2}, \mathbf{D}_{\geq 2} + \mathbf{Z}_{\geq 2}, \mathbf{C}_{\geq 2}),$$

for some $\mathbf{Z} \in \mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C})$.

As $\hat{\mathbf{Z}} \in \mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C})$, *consistency with direct transfer* is a stronger condition than *consistency*. Still, for some problems the two axioms have the same implication: for instance, if $\mathbf{X}$ is such that agent 2 takes on all tasks left behind by agent 1, then the only reduced problem is as in *consistency with direct transfer*. However, in general, there will be an infinite number of valid reduced problems, each possibly inducing different recognition probabilities.

Example 2. We continue on Example 1 and explore all ways $\mathbf{Z} \in \mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C})$ to distribute agent 1's excess $e_1 - x_1 = (4, 0, 0, 0, 0)$. Thus, it suffices to consider type $t = 1$ for which, in addition, no agent $j \geq 2$ has any contracts initially, $d_{j1} = 0$.

1. For agent 1, we set $z_{11} = 0$.
2. Agent 2 should be able to complete their announced tasks, $z_{21} \geq x_{21} = 1$, but not hold more than their capacity, $z_{21} \leq c_{21} = 2$.
3. Agent 3 may indirectly receive some excess through agent 2 if $z_{21} > x_{21}$. Still, together they should have enough to complete their tasks, $z_{21} + z_{31} \geq x_{21} + x_{31} = 2$, while agent 3 should not go beyond capacity, $z_{31} \leq c_{31} = 2$.
4. We derive similar bounds for agent 4: $z_{21} + z_{31} + z_{41} \geq x_{21} + x_{31} + x_{41} = 3$ and $z_{41} \leq c_{41} = 1$.
5. Finally, for agent 5, $z_{11} + \cdots + z_{51} = d_{11} - x_{11} = 4$ yields $z_{51} = 4 - z_{21} - z_{31} - z_{41}$.

The possible assignments to later agents depend on how much earlier agents are given: for instance, if $z_{21} = 1$, then $1 \leq z_{31} \leq 2$; on the other hand, if $z_{21} = 2$, then $0 \leq z_{31} \leq 2$. In Figure 2, the possible distributions among agents 3, 4, and 5 given that agent 2 are assigned $z_{21} = 2$ is illustrated through the light and dark gray areas. ○

3.4 Top-agent proportionality

In Section 4, Proposition 2 will show that an implication of *cost-effective implementation* is that agent 1's recognition probability cannot depend on the announcements $\mathbf{X}$ at all. That is to say, if p_1 were to depend on x_1, then agent 1 could be better off announcing an inefficient workload. Likewise, if p_1 were to depend on some other x_i, then agent 1 could, through their announcement, affect the tasks available to later agents and thereby affect agent i's announcement and, in effect, affect 1's own recognition probability. In this way, agent 1 could again be better off announcing an inefficient workload. Thus, insisting on *cost-effective implementation*, agent 1's probability can essentially depend only on their contracted tasks d_1. *Top-agent proportionality* asserts that agent 1's probability should equal her fraction of the contracted tasks.

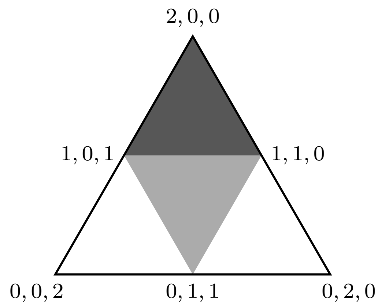

Figure 2: The light and dark gray areas together give all ways to share agent 1's excess among agents 3, 4, and 5 given that agent 2 is assigned $z_{21} = 2$, so $\{(z_{31}, z_{41}, z_{51}) : \mathbf{Z} \in \mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C}), z_{21} = 2\}$. The dark gray area is the set $\{(z_{31}, z_{41}, z_{51}) : \mathbf{Z} \in \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})\}$ referred to in Example 3. The white areas are infeasible due to capacity constraints.

Axiom 3 (Top-agent proportionality)**.** For each population size $n \in \mathbb{N}$ and problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$,

$$p_1(\mathbf{X}, \mathbf{D}, \mathbf{C}) = \frac{\sum_t d_{1t}}{\sum_i \sum_t d_{it}}.$$

For our practical applications, we envision task quantities to represent iterations, computation time, or similar aspects. Thus, just as when dealing with the cost function, we are justified in adding quantities of different task types.

4 Results

We now turn to our formal results. First, we take as given a particular two-stage mechanism (as represented by an arbitrary recognition function p) and analyze its subgame-perfect equilibria through backward induction. Thus, we first solve for the equilibrium bargaining behavior in the second stage of the mechanism. Proposition 1 shows that, in equilibrium, the cost savings $\Delta(\mathbf{X}, \mathbf{D}, f)$ will in expectation be shared in proportion to the recognition probabilities $p_i(\mathbf{X}, \mathbf{D}, \mathbf{C})$. Thereafter, we proceed with the axiomatic analysis and pin down the class of recognition functions that satisfy *cost-effective implementation*: Proposition 2 shows that these functions are such that each agent i's probability p_i is independent of the announcements $x_i, \ldots, x_n$. Thereafter, Proposition 3 turns to our other axioms to establish the functional form of all recognition functions that satisfy *consistency* and *top-agent proportionality*. Proposition 4 connects the prior results, giving a complete description of all reduced problems with respect to which the recognition function can be *consistent* yet still satisfy *cost-effective implementation* and *top-agent proportionality*. Finally, Theorem 1 shows that the three axioms pin down a particular recognition function p^*. Expressed differently, the two classes identified in Propositions 2 and 3, respectively, overlap in a single recognition function p^*. In this way, Theorem 1 provides a normative foundation for p^*, a recognition function which guarantees that agents announce cost-effective workloads all the while ensuring that the cost savings are shared fairly.

4.1 Equilibrium behavior in the bargaining stage

Fix a matrix $\mathbf{D} \in \tilde{\mathbb{D}}$ together with a capacity matrix $\mathbf{C} \geq \mathbf{D}$ and cost functions f. We analyze the two-stage mechanism from the end, starting with its second stage. Hence, take as given a first-stage announcement $\mathbf{X}$.

Recall that agents bargain on how to share the costs $F(\mathbf{X})$, or, equivalently, on which solution in $U(\mathbf{X})$ to implement. Recall also that we restrict to stationary strategies in which each agent i always proposes the same $u^i \in U(\mathbf{X})$ and always accepts any proposal that awards i a net cost of at most $a_i \in \mathbb{R}$.

Proposition 1 shows that, in the subgame-perfect equilibrium of the bargaining stage, agents share the cost savings in expectation in proportion to the recognition probabilities. That is, all agents pay $f_i(d_i)$ for the tasks that they have contracted on, but, on top of that, save a share $p_i(\mathbf{X}, \mathbf{D}, \mathbf{C})$ of the total savings $\Delta(\mathbf{X}, \mathbf{D}, f)$. In this way, the expected payoffs are independent of the continuation probability δ while the equilibrium strategies depend on δ in a natural way: the greater the probability of the bargaining process continuing, the more the proposer has to award the other agents in order to reach an agreement (compare Gudmundsson et al., 2019, Figure 1). A general discussion of the uniqueness of equilibrium payoffs in coalitional bargaining models can be found in Eraslan and McLennan (2013) and references therein.

Proposition 1. *An agent's expected cost is the same in all stationary, subgame-perfect equilibria of Stage 2. In particular,*

A. If $\Delta(\mathbf{X}, \mathbf{D}, f) \leq 0$, then agent i's expected cost is $f_i(d_i)$, and there are several cost-equivalent equilibria;

B. If $\Delta(\mathbf{X}, \mathbf{D}, f) > 0$, then agent i's expected cost is $f_i(d_i) - p_i(\mathbf{X}, \mathbf{D}, \mathbf{C}) \cdot \Delta(\mathbf{X}, \mathbf{D}, f)$, and the equilibrium is unique.

All proofs are postponed to the Appendix. Next, we turn to analyzing the game as a whole.

4.2 The implication of cost-effective implementation

Agents foresee that their equilibrium payoffs in the second stage will be as described in Proposition 1: they are awarded a share of the total savings that is proportional to their recognition probability. For that reason, when announcing their desired workloads in the first stage, they may face a trade-off: if they can announce an inefficient amount that reduces total savings but sufficiently increases their own probability, they may be better off. Of course, if an agent cannot affect their own recognition probability through their announcement, then they have nothing to gain from inefficient announcements. Thus, recognition functions that prevent agents from affecting their probabilities satisfy *cost-effective implementation.* Indeed, Proposition 2 shows that these are the only recognition functions to do so.

Proposition 2. *The recognition function p satisfies cost-effective implementation if and only if, for each matrix $\mathbf{D} \in \tilde{\mathbb{D}}$, capacity matrix $\mathbf{C}$, agent j, and pair of allocations $\mathbf{X}, \mathbf{Y} \in \mathcal{X}(\mathbf{D}, \mathbf{C})$ that coincide for all agents $i < j$, agent j's recognition probability is the same:*

$$\mathbf{X}_{<j} = \mathbf{Y}_{<j} \implies p_j(\mathbf{X}, \mathbf{D}, \mathbf{C}) = p_j(\mathbf{Y}, \mathbf{D}, \mathbf{C}).$$

A first implication of Proposition 2 is that agent 1's recognition probability p_1 is independent of the announcements $\mathbf{X}$; this has been used as an argument in favor of *top-agent proportionality.* A second implication is that agent j's recognition probability p_j cannot depend on j's own announcement x_j; this will later be used to pin down the *consistent* and *top-agent proportional* recognition functions that satisfy *cost-effective implementation.*

4.3 The class of consistent and top-agent proportional recognition functions

Next, we identify the class of recognition functions characterized jointly by *consistency* and *top-agent proportionality.* The latter axiom immediately settles agent 1's probability (below, $p_1(\mathbf{X},\mathbf{D},\mathbf{C}) = g(1)$). At that point, we can appeal to *consistency* to reduce the problem and distribute 1's excess through some $\mathbf{Z}^1 \in \mathbb{Z}(\mathbf{X},\mathbf{D},\mathbf{C})$. Through *top-agent proportionality*, we can then pin down the probability for the "new" first agent in the reduced problem with $n-1$ agents. In this way, going back and forth between the axioms, distributing each agent j's excess through some $\mathbf{Z}^j \in \mathbb{Z}(\mathbf{X}_{\geq j}, \mathbf{D}_{\geq j} + \mathbf{Z}^1_{\geq j} + \cdots + \mathbf{Z}^{j-1}_{\geq j}, \mathbf{C}_{\geq j})$, we derive a particular member of the characterized class.[9]

Proposition 3. *The recognition function p satisfies consistency and top-agent proportionality if and only if, for each population size $n \in \mathbb{N}$ and problem $(\mathbf{X},\mathbf{D},\mathbf{C}) \in \mathcal{P}^n$, there exist $\mathbf{Z}^1, \dots, \mathbf{Z}^{n-1}$ such that $\mathbf{Z}^1 \in \mathbb{Z}(\mathbf{X},\mathbf{D},\mathbf{C})$ and, for each $j \geq 2$, $\mathbf{Z}^j \in \mathbb{Z}(\mathbf{X}_{\geq j}, \mathbf{D}_{\geq j} + \mathbf{Z}^1_{\geq j} + \cdots + \mathbf{Z}^{j-1}_{\geq j}, \mathbf{C}_{\geq j})$, such that*

$$p_1(\mathbf{X},\mathbf{D},\mathbf{C}) = g(1) = \frac{\sum_t d_{1t}}{\sum_i \sum_t d_{it}}$$

and, for each agent $j \geq 2$,

$$p_j(\mathbf{X},\mathbf{D},\mathbf{C}) = (1-g(1)) \cdots (1-g(j-1)) \cdot g(j),$$

where, for each $j \geq 2$,

$$g(j) = \frac{\sum_t d_{jt} + \sum_{i<j} \sum_t z^i_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j} \sum_t x_{it}}.$$

To summarize, Proposition 2 provides a precise description of the class of recognition functions that satisfy *cost-effective implementation*, while Proposition 3 pins down the functional form of recognition functions satisfying *consistency* and *top-agent proportionality.* Next, we turn to the intersection of these classes.

4.4 All forms of consistency compatible with cost-effective implementation and top-agent proportionality

We now determine all forms of consistency that are compatible with *cost-effective implementation* and *top-agent proportionality*; that is, we find all ways to specify reduced problems such that the axioms are compatible. These distributions of the first agent's excess form a sort of "equivalence class" $\mathbb{Z}^*(\mathbf{X},\mathbf{D},\mathbf{C}) \subseteq \mathbb{Z}(\mathbf{X},\mathbf{D},\mathbf{C})$ within the set of distributions. The class includes $\hat{\mathbf{Z}}$ as defined in *consistency with direct transfer* as an extreme member, but there are generally many other distributions as well.

We now outline an arbitrary distribution $\mathbf{Z} \in \mathbb{Z}^*(\mathbf{X},\mathbf{D},\mathbf{C})$; see Appendix A for additional arguments verifying that $\mathbf{Z}$ always is well-defined. The construction borrows from *consistency with direct transfer*, differing only in Case 2 below. We set $z_{1t} = 0$ and then determine $z_{2t}, z_{3t}, \dots$ recursively. In general, for each agent $j = 2, 3, \dots$, we check whether $\sum_{i \leq j} d_{it} \leq \sum_{i<j} x_{it} + c_{jt}$:

Case 1. If so, we set $z_{jt} \geq 0$ such that $\sum_{i \leq j} z_{it} = d_{1t} - x_{1t}$ and $z_{j+1,t} = \cdots = z_{nt} = 0$, building on the values z_{it} for $i < j$ determined in previous steps.

[9] Indices refer to agents rather than matrix rows: z^i_j is agent j's assignment of agent i's excess, so row $j-i$ of $\mathbf{Z}^i \in \mathbb{R}^{(n-i+1)\times m}$.

CASE 2. Otherwise, we set $0 \leq z_{jt} \leq c_{jt} - d_{jt}$ such that

$$\sum_{i\leq j}(d_{it} + z_{it}) \geq \sum_{i<j} x_{it} + c_{jt} \tag{$\star$}$$

and continue to the next agent.

Compared to $\hat{\mathbf{Z}}$, we find that z_{2t} coincides with $\hat{z}_{2t}$ whereas later values z_{jt} may differ. However, if we use the upper bound $z_{jt} = c_{jt} - d_{jt}$ whenever in CASE 2, then we obtain $\mathbf{Z} = \hat{\mathbf{Z}}$, so $\hat{\mathbf{Z}} \in \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$. On the other hand, the lower bound is interpreted as follows. On top of what agents 1 through j have contracted on, $\sum_{i\leq j} d_{it}$, they are assigned sufficiently many tasks, $\sum_{i\leq j} z_{it}$, to meet the actual announcements by all but agent j, $\sum_{i<j} x_{it}$, together with the maximal announcement $\bar{x}_{jt}$ by agent j.

Example 3. Continuing on Examples 1 and 2, we turn to the distributions $\mathbf{Z} \in \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$. For agent 2, as $d_{11} + d_{21} \nleq x_{11} + c_{21}$, we set $z_{21} = 2$. This restricts the possible distributions compared to $\mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C})$. Next, as $d_{11} + d_{21} + d_{31} \nleq x_{11} + x_{21} + c_{31}$, we set $0 \leq z_{31} \leq 2$ such that $d_{21} + d_{31} + z_{21} + z_{31} \geq x_{21} + c_{31}$, which reduces to $1 \leq z_{31} \leq 2$. This imposes further restrictions on the set of distributions compared to $\mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C})$. Any remaining excess, $4 - z_{21} - z_{31} = 2 - z_{31} \leq 1$, can be distributed in any way between agents 4 and 5. In Figure 2, $\mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$ corresponds to the dark gray triangle. ∘

The next result characterizes the *consistent* and *top-agent proportional* recognition functions p that satisfy *cost-effective implementation*. In particular, p must be consistent with respect to $\mathbb{Z}^*(\cdot)$. That is, if p ever is consistent with respect to some $\mathbf{Z}' \notin \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$, then some agent can be better off announcing an inefficient workload. On the other hand, if p always is consistent with respect to some $\mathbf{Z} \in \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$, then p is consistent with respect to all matrices in $\mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$ and satisfies *cost-effective implementation.*

Proposition 4. *Let p satisfy consistency and top-agent proportionality. Then p satisfies cost-effective implementation if and only if, for each population size $n \in \mathbb{N}$ and problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \tilde{\mathcal{P}}^n$, p is consistent with respect to some $\mathbf{Z} \in \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$ and no $\mathbf{Z}' \notin \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$.*

Next, we will see that there is only one recognition function that satisfies the three axioms. That is to say, regardless with respect to which $\mathbf{Z} \in \mathbb{Z}^*(\mathbf{X}, \mathbf{D}, \mathbf{C})$ the function is consistent, the resulting probabilities are the same, confirming the idea that $\mathbb{Z}^*$ forms an "equivalence class" with $\mathbb{Z}$.

4.5 Main characterization

Theorem 1 shows that the three axioms jointly single out a unique recognition function p^*. This function essentially splits the recognition probability in proportion to the contracts available to the agent when it is their turn to announce their desired workload. That is, agent 1's probability equals her share of the total contracts (by *top-agent proportionality*); next, all of 1's excess that agent 2 can hold is added to 2's contracts when computing 2's share of the remaining tasks; and so on.

Theorem 1. *The recognition function p satisfies cost-effective implementation, consistency, and top-agent proportionality if and only if, for each population size $n \in \mathbb{N}$ and problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \tilde{\mathcal{P}}^n$,*

$$p_1(\mathbf{X}, \mathbf{D}, \mathbf{C}) = p_1^*(\mathbf{X}, \mathbf{D}, \mathbf{C}) = g(1) = \frac{\sum_t d_{1t}}{\sum_i \sum_t d_{it}}$$

and, for each agent $j \geq 2$,

$$p_j(\mathbf{X},\mathbf{D},\mathbf{C}) = p_j^*(\mathbf{X},\mathbf{D},\mathbf{C}) = (1-g(1))\cdots(1-g(j-1))\cdot g(j),$$

where, for each $j \geq 2$,

$$g(j) = \frac{\sum_t d_{jt} + \sum_{i<j}\sum_t \hat{z}^i_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j}\sum_t x_{it}} = \frac{\sum_t \bar{x}_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j}\sum_t x_{it}} = \frac{\sum_t \min\{\sum_{i\leq j} d_i - \sum_{i<j} x_i, c_j\}_t}{\sum_i \sum_t d_{it} - \sum_{i<j}\sum_t x_{it}}.$$

The statement gives different ways to describe the function g. From Proposition 3, we know the functional form under *consistency* and *top-agent proportionality*; here, Theorem 1 shows that when also *cost effective implementation* is satisfied, the function g is as in the case where the excess-distribution matrices $\mathbf{Z}$ coincide with $\hat{\mathbf{Z}}$ as in *consistency with direct transfer*. Moreover, the probabilities are again determined recursively, starting from "the top". The share agent j gets out of the remaining probability mass $1-(p_1^*+\cdots+p_{j-1}^*)$ is given by the size of their maximal announcement $\bar{x}_j$ in relation to the total remaining tasks $\sum_i d_i - \sum_{i<j} x_i$. As agent j's probability is independent of the announced workloads by those more complex than j, the recognition function p^* depends on the announcements while still satisfying *cost-effective implementation.*

Example 4. We continue on Examples 1 through 3 to compute the maximal announcements using

$$\bar{x}_{jt} = \min\{\textstyle\sum_{i\leq j} d_{it} - \sum_{i<j} x_{it}, c_{jt}\}.$$

Specifically, $\bar{\mathbf{X}}$ coincides with $\mathbf{C}$ except $\bar{x}_{42} = \bar{x}_{53} = 1$ and $\bar{x}_{52} = 0$.[10] The probabilities p^* are as follows:

$$\bar{\mathbf{X}} = \begin{bmatrix} 9 & 0 & 0 & 0 & 0 \\ 2 & 5 & 0 & 0 & 0 \\ 2 & 2 & 4 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{aligned} p_1^* &= 9/21 = 3/7 \\ p_2^* &= (7/16)\cdot(1-3/7) = 1/4 \\ p_3^* &= (8/12)\cdot(1-3/7-1/4) = 3/14 \\ p_4^* &= (5/8)\cdot(1-3/7-1/4-3/14) = 15/224 \\ p_5^* &= 1-3/7-1/4-3/14-15/224 = 9/224. \end{aligned}$$

Expressed differently, $p^*(\mathbf{X},\mathbf{D},\mathbf{C}) = (96,56,48,15,9)/224 \approx (.43,.25,.21,.07,.04)$. As an example, with identical cost functions $f_j(z) = z^2$, the costs are reduced from $(81,25,16,4,1)$ at $\mathbf{D}$ through the total savings $\Delta(\mathbf{X},\mathbf{D},f) = 38$ to approximately $(64.7, 15.5, 7.9, 1.4, -0.5)$ in expectation in equilibrium. ∘

Next, we show that the axioms imposed in Theorem 1 are independent. To do so, we design recognition functions that satisfy two of the properties but not the third. First, the recognition function p such that $p_j(\mathbf{X},\mathbf{D},\mathbf{C}) = 1/n$ for each agent j satisfies *cost-effective implementation* and *consistency* but not *top-agent proportionality.* Second, if we set probabilities proportional to $\mathbf{D}$,

$$\tilde{p}_j(\mathbf{X},\mathbf{D},\mathbf{C}) = \frac{\sum_t d_{jt}}{\sum_i \sum_t d_{it}},$$

then $\tilde{p}$ satisfies *cost-effective implementation* and *top-agent proportionality.* However, $\tilde{p}$ fails *consistency* for the problem $(\mathbf{X},\mathbf{D},\mathbf{C})$ with non-zero entries $d_{11} = 3$, $d_{22} = 2$, $d_{33} = 1$; $\mathbf{C} = \mathbf{D}$ except $c_{31} = 1$; and $\mathbf{X} = \mathbf{C}$ except $x_{11} = 2$. Once agent 1's probability is settled, $\tilde{p}_1(\mathbf{X},\mathbf{D},\mathbf{C}) = 3/6$, there is only one way $\tilde{\mathbf{Z}}$ to distribute

[10] The corresponding maximal announcements given $\mathbf{Y}$ differ from $\bar{\mathbf{X}}$ through $\bar{y}_4 = (0,2,1,2,0)$ and $\bar{y}_5 = (0,1,2,0,1)$. This difference is of no consequence for the probabilities as $\sum_t \bar{y}_{jt} = \sum_t \bar{x}_{jt}$ for each agent j.

1's excess, so $\mathbb{Z}(\mathbf{X},\mathbf{D},\mathbf{C}) = \mathbb{Z}^*(\mathbf{X},\mathbf{D},\mathbf{C}) = \{\tilde{\mathbf{Z}}\}$, where $\tilde{z}_{31} = 1$ and $\tilde{z}_{it} = 0$ otherwise. Then *consistency* and *top-agent proportionality* would require agent 2's probability to be $(2/4) \cdot (1 - 3/6) \neq 2/6 = \tilde{p}_2(\mathbf{X},\mathbf{D},\mathbf{C})$.

To satisfy *consistency* and *top-agent proportionality* but not *cost-effective implementation*, we first appeal to Proposition 3. Borrowing parts of Proposition 3, the recognition function is defined through some matrices $\mathbf{Z}^1, \dots, \mathbf{Z}^{n-1}$ and the function g such that, for agent $j \geq 2$,

$$g(j) = \frac{\sum_t d_{jt} + \sum_{i<j} \sum_t z^i_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j} \sum_t x_{it}}.$$

For p^*, each matrix $\mathbf{Z}^j$ is given by the corresponding $\hat{\mathbf{Z}}$ as in *consistency with direct transfer*. This assigns as much as possible of the excess in order $2, 3, \dots$: agent 2 gets as much as possible, then agent 3 gets as much as possible out of what remains, and so on. Different orders induce different recognition functions. In particular, for the function $\check{p}$, we use the order $2, \dots, n-2, n, n-1$. Specifically, $\check{p}$ is defined around $\mathbf{Z}$ with $z_n = \min\{\hat{z}_{n-1} + \hat{z}_n, x_n - e_n\}$ and $z_{n-1} = \hat{z}_{n-1} + \hat{z}_n - z_n$. As $\hat{\mathbf{Z}}$ is well-defined and $z_{n-1} + z_n = \hat{z}_{n-1} + \hat{z}_n$, $0 \leq z_{n-1} \leq \hat{z}_{n-1}$, and $e_n + z_n \leq x_n$, so is $\mathbf{Z}$.

To show that $\check{p}$ fails *cost-effective implementation*, modify the problem $(\mathbf{X},\mathbf{D},\mathbf{C})$ above through capacities $\check{\mathbf{C}} = \mathbf{C}$ except $\check{c}_{21} = 1$. Then $\mathbf{Z} \in \mathbb{Z}(\mathbf{X},\mathbf{D},\check{\mathbf{C}})$ whenever $z_{21} + z_{31} = 1$, $z_{21}, z_{31} \geq 0$, and otherwise $z_{it} = 0$. We construct $\check{p}$ around $\tilde{\mathbf{Z}}$ defined above, so $\tilde{z}_{31} = 1$. In particular, we have $\check{p}_2(\mathbf{X},\mathbf{D},\check{\mathbf{C}}) = (2/4) \cdot (1 - 3/6)$. Suppose now instead that agent 2 announces $\check{x}_2 = \bar{x}_2 = (1, 2, 0)$, so we end on $\check{\mathbf{X}}$ such that $\check{\mathbf{X}} = \mathbf{X}$ except $\check{x}_{21} = 1$ and $\check{x}_{31} = 0$. Then $\mathbb{Z}(\check{\mathbf{X}},\mathbf{D},\check{\mathbf{C}}) = \{\check{\mathbf{Z}}\}$, where $\check{z}_{21} = 1$ and $\check{z}_{it} = 0$ otherwise. By *consistency* and *top-agent proportionality*, $\check{p}_2(\check{\mathbf{X}},\mathbf{D},\check{\mathbf{C}}) = (3/4) \cdot (1 - 3/6) > \check{p}_2(\mathbf{X},\mathbf{D},\check{\mathbf{C}})$. In particular, there are costs f for which $\mathbf{X}$ is cost effective yet agent 2 is better off at $\check{\mathbf{X}}$ than at $\mathbf{X}$.

5 Concluding remarks

The intrinsically decentralized nature of the new, digital economy brings about new challenges, one of which is the construction of institutions to align individual incentives with collective, societal objectives (compare Chen et al., 2020). We have taken one step towards designing mechanisms to support efficient collaboration between independent agents, yet there remain many interesting avenues for future research.

Throughout, we have assumed costs to be common knowledge. While this can be challenged in practice, we still contend that agents with privately held costs, through repeated interaction, would get better informed on each other's costs and eventually converge on desirable allocations supported by agreeable payments. Related, we have not explored the computational aspect of the agents' optimization problem. However, in the present model this is a straightforward minimization of a convex function (the total costs). In practice, the optimization problem may be more involved. For instance, tasks may not be divisible. While this would add new constraints to the reallocation possibilities, it would not require a significant change to the model. Alternatively, capacities may not be separable in tasks as assumed here. For instance, different "weights" may be assigned different tasks, and agents may be limited in the total "weight" of their assignment. Again, the optimization problem gets more intricate, but our solution generalizes readily once we account for the more complex computation needed to find the agents' maximum announcements. Finally, in practice, we might expect agents' capacities to be expressed more through the cost functions than through hard constraints. However, this can be viewed as a special case of the model, covered by setting capacities to $c_{it} = 0$ for feasibility constraints and otherwise $c_{it} = K$ for a large enough K. A more extensive modification of the model that we have not considered is to model time, say arrival and completion times of tasks. Lastly, we "step in" at a point in which agents already hold some initially-signed contracts. An interesting avenue for future research is to generalize this and model also the interaction with the task providers. An intermediate step towards this may be to study a model in which agents may choose not to complete some tasks.

Thus far, we have not discussed *who* chooses the mechanism but rather argued in favor of a particular mechanism. Again, it is intrinsic to the open, blockchain-based economy that *anyone* can set up a smart contract. This leaves little room for rent extraction for the contract creator as anyone can copy the contract, make modifications, and invite the service providers to settle their allocation on the alternative platform. Thus, we expect that whether a mechanism will thrive in practice will depend on the fairness of the solution it provides. The reasoning is entirely analogous to what has been observed for centralized matching mechanisms, which, as noted for instance by Roth (1991), are most often successful when the outcomes they produce are perceived as fair ("stable"). Hence, similar to the evolution of norms in society, we expect to converge towards a mechanism that ensures cost-effective allocations with fair sharing of the cost reductions.

References

Abadi, J. and M. Brunnermeier (2018). Blockchain Economics. *NBER Working Paper No. 25407.*

Azar, Y., I. Kalp-Shaltiel, B. Lucier, I. Menache, J. S. Naor, and J. Yaniv (2015). Truthful online scheduling with commitments. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, New York, NY, USA, pp. 715–732. Association for Computing Machinery.

Bahel, E. and C. Trudeau (2019). Stability and fairness in the job scheduling problem. *Games and Economic Behavior 117*(C), 1–14.

Bao, Y., Y. Peng, C. Wu, and Z. Li (2018). Online job scheduling in distributed machine learning clusters. *CoRR abs/1801.00936.*

Baron, D. P. and J. A. Ferejohn (1989). Bargaining in legislatures. *The American Political Science Review 83*(4), 1181–1206.

Bertsimas, D. and V. F. Farias (2012). On the Efficiency-Fairness Trade-off. *Management Science 58*(12), 2234–2250.

Binmore, K. (1987). *Perfect equilibria in bargaining models. In: Binmore K., Dasgupta P. (Eds.) The Economics of Bargaining*, Chapter 5. Blackwell, Oxford.

Binmore, K., A. Rubinstein, and A. Wolinsky (1986). The Nash Bargaining Solution in Economic Modelling. *The RAND Journal of Economics 17*(2), 176–188.

Brucker, P. (2004). *Scheduling Algorithms,* (fourth ed.). Springer.

Cachon, G. P. and M. A. Lariviere (1999). Capacity Choice and Allocation: Strategic Behavior and Supply Chain Performance. *Management Science 45*(8), 1091–1108.

Can, B. (2019). Economic Design of Things. In Laslier, Moulin, Sanver, and Zwicker (Eds.), *The Future of Economic Design*, pp. 487–493. Springer International Publishing.

Catalini, C. and J. S. Gans (2020). Some Simple Economics of the Blockchain. *Communications of the ACM 63*(7), 80–90.

Chen, L., S. Liu, B. Li, and B. Li (2019). Scheduling jobs across geo-distributed datacenters with max-min fairness. *IEEE Transactions on Network Science and Engineering 6*(3), 488–500.

Chen, Y., P. Cramton, J. A. List, and A. Ockenfels (2020). Market Design, Human Behavior, and Management. *Management Science*.

Chun, Y. (2016). *Fair Queueing*. Springer, New York, NY.

Corchón, L. C. (2009). Implementation Theory. In R. Meyers (Ed.), *Encyclopedia of Complexity and Systems Science*. Springer, New York, NY.

Dutta, A., G. Koehler, and A. Whinston (1982). On optimal allocation in a distributed processing environment. *Management Science 28*(8), 839–853.

Eraslan, H. and A. McLennan (2013). Uniqueness of stationary equilibrium payoffs in coalitional bargaining. *Journal of Economic Theory 148*(6), 2195 – 2222.

Ernst, A., H. Jiang, and M. Krishnamoorthy (2006). Exact solutions to task allocation problems. *Management Science 52*(10), 1634–1646.

Gans, J. (2019). The fine print in smart contracts. *NBER Working Paper 25443*.

Gudmundsson, J., J. L. Hougaard, and C. Y. Ko (2019). Decentralized mechanisms for river sharing. *Journal of Environmental Economics and Management 94*, 67 – 81.

Hillier, F. and G. Lieberman (2021). *Introduction to Operations Research* (11 ed.). McGraw Hill.

Holmstrom, B. (1982). Moral Hazard in Teams. *The Bell Journal of Economics 13*(2), 324–340.

Hu, X., R. Caldentey, and G. Vulcano (2013). Revenue Sharing in Airline Alliances. *Management Science 59*(5), 1177–1195.

Huang, Z., B. Balasubramanian, M. Wang, T. Lan, M. Chiang, and D. Tsang (2015, August). Need for speed: Cora scheduler for optimizing completion-times in the cloud. In *2015 IEEE Conference on Computer Communications, IEEE INFOCOM 2015*, Proceedings - IEEE INFOCOM, United States, pp. 891–899. Institute of Electrical and Electronics Engineers Inc.

Kouvelis, P. and M. A. Lariviere (2000). Decentralizing Cross-Functional Decisions: Coordination Through Internal Markets. *Management Science 46*(8), 1049–1058.

Li, Z., J. Ge, H. Yang, L. Huang, H. Hu, H. Hu, and B. Luo (2016, December). A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. *Future Generation Computer Systems 65*, 140–152.

Maskin, E. S. and T. Sjöström (2002). Implementation Theory. In KJ Arrow and AK Sen (Ed.), *Handbook of Social Choice and Welfare*, Chapter 5. Elsevier, Amsterdam.

Min, T., H. Wang, Y. Guo, and W. Cai (2019). Blockchain Games: A Survey. In *2019 IEEE Conference on Games (CoG)*, pp. 1–8.

Moulin, H. (2007). On scheduling fees to prevent merging, splitting and transferring of jobs. *Mathematics of Operations Research 32*(2), 266–283.

Roth, A. E. (1991). A natural experiment in the organization of entry-level labor markets: Regional markets for new physicians and surgeons in the United Kingdom. *American Economic Review 81*, 415–440.

Tawarmalani, M., K. Kannan, and P. De (2009). Allocating Objects in a Network of Caches: Centralized and Decentralized Analyses. *Management Science 55*(1), 132–147.

Thomson, W. (2011). Consistency and its converse: an introduction. *Review of Economic Design 15*(4), 257–291.

Thomson, W. (2016). Fair allocation. In M. D. Adler and M. Fleurbaey (Eds.), *The Oxford Handbook of Well-Being and Public Policy*, Chapter 7. Oxford University Press.

Thomson, W. (2019). On the Axiomatics of Resource Allocation: Classifying Axioms and Mapping Out Promising Directions. In Laslier, Moulin, Sanver, and Zwicker (Eds.), *The Future of Economic Design*, pp. 213–222. Springer International Publishing.

World Bank Group (2020). Smart Contract Technology and Financial Inclusion. *FinTech Note No. 6*.

Zhang, X., Z. Huang, C. Wu, Z. Li, and F. C. Lau (2015). Online auctions in iaas clouds: Welfare and profit maximization with server costs. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '15, New York, NY, USA, pp. 3–15. Association for Computing Machinery.

Zhou, R., Z. Li, C. Wu, and Z. Huang (2017, April). An efficient cloud market mechanism for computing jobs with soft deadlines. *IEEE/ACM Trans. Netw. 25*(2), 793–805.

A Non-emptiness of $\mathbb{Z}^*$

For convenience, we drop the subscript t throughout. If we reach step j in the construction of $\mathbf{Z}$, then we must have gone through Case 2 for agent $j-1$. By equation $\star$ for $j-1$ and as $c_{j-1} \geq x_{j-1}$,

$$\sum_{1<i<j}(d_i + z_i) \geq \sum_{1<i<j-1} x_i + c_{j-1} \geq \sum_{1<i<j} x_i \implies \sum_{1<i<j}(d_i + z_i - x_i) \geq 0.$$

Suppose Case 1 applies to agent j. We will verify that $d_j + z_j \leq c_j$. By Case 1, $\sum_{1<i\leq j} z_i = d_1 - x_1$ and

$$c_j \geq \sum_{i\leq j} d_i - \sum_{i<j} x_i = \sum_{1<i\leq j}(d_i + z_i) - \sum_{1<i<j} x_i = d_j + z_j + \sum_{1<i<j}(d_i + z_i - x_i) \geq d_j + z_j.$$

Suppose instead Case 2 applies to agent j. We will verify that equation $\star$ can be satisfied for some $0 \leq z_j \leq c_j - d_j$. In particular, it is true for $z_j = c_j - d_j$ as then

$$\sum_{1<i\leq j}(d_i + z_i) = \sum_{1<i<j}(d_i + z_i) + c_j \geq \sum_{1<i<j} x_i + c_j.$$

Here, again, we appeal to equation $\star$ for agent $j-1$. Finally, if we reach agent n, then $\sum_i d_i = \sum_i x_i \leq \sum_{i<n} x_i + c_n$, so we fall into Case 1, finalizing the construction of $\mathbf{Z}$.

B Proofs

When it causes no confusion, we ease the notation as follows in the proofs: $p_i(\mathbf{X}, \mathbf{D}, \mathbf{C})$ is referred to as $p_i(\mathbf{X})$, $\Delta(\mathbf{X}, \mathbf{D}, f)$ as $\Delta(\mathbf{X})$, and when tasks can be dealt with independently, we sometimes drop the subscript t. The proofs of Proposition 1 and 2 mirror proofs found in Gudmundsson et al. (2019).

B.1 Proof of Proposition 1

Let $f_j^0 \equiv f_j(\sum_t d_{jt})$ denote agent j's cost at $\mathbf{D}$. If j sets their acceptance threshold a_j to f_j^0, then either agreement is reached and j pays at most f_j^0, or the negotiations fail, we revert to $\mathbf{D}$, and j pays f_j^0. In this way, there cannot be an equilibrium in which j pays more than f_j^0. Moreover, $a_j \le f_j^0$ in equilibrium.

(Part A) If $\Delta(\mathbf{X}) < 0$, then there is no proposal $u \in U(\mathbf{X})$ such that $u \le f^0$, so agreement cannot be reached, and thus expected costs are $\mathbb{E}\pi = f^0$ (that is, $\mathbb{E}\pi_j = f_j^0$ for each agent j). If $\Delta(\mathbf{X}) = 0$, then only the proposal $u = f^0$ can be agreed on and, again, expected cost are $\mathbb{E}\pi = f^0$. In these cases, we have $a = f^0$ in equilibrium (recall, we restrict to subgame-perfect equilibria). To see this, suppose for contradiction that there is an equilibrium with $a_j < f_j^0$ for some agent j and in which agreement never is reached. Then there is a subgame off the equilibrium path in which all agents but j accept the proposal and the offer to j, u_j, is such that $a_j < u_j < f_j^0$. Then j is better off accepting the proposal (cost u_j) rather than rejecting it (cost f_j^0), a contradiction to it being an equilibrium.

(Part B) Next, assume that $\Delta(\mathbf{X}) > 0$, so there is $u \in U(\mathbf{X})$ such that $u \le f^0$. We claim that agreement will be reached in equilibrium. For contradiction, suppose agreement is not reached. Then equilibrium expected costs and acceptance thresholds are $\mathbb{E}\pi = a = f^0$. Every agent j is then better off proposing, for instance, the costs u such that $u_i = f_i^0 - p_i(\mathbf{X}) \cdot \Delta(\mathbf{X}) \le a_i$ for each agent i. This proposal would be accepted and decrease the proposer's expected costs, a contradiction to it being an equilibrium. Thus, agreement will be reached in equilibrium. Moreover, no proposal is ever rejected: if, say, agent j were to make a proposal that was rejected while, among the agents who make accepted proposals, agent k makes the proposal that assigns the lowest cost to j, then j is better off offering k's proposal. In this way, agreement is always reached in the first round of negotiations.

Let u_j^i denote the cost that agent i proposes agent j to bear. Then the expected cost of agent j is $\mathbb{E}\pi_j = \sum_i p_i(\mathbf{X}) \cdot u_j^i$. As each proposal u^i is in $U(\mathbf{X})$, $\sum_j u_j^i = F(\mathbf{X})$, and as $\sum_i p_i(\mathbf{X}) = 1$,

$$\textstyle\sum_j \mathbb{E}\pi_j = \sum_i \sum_j p_i(\mathbf{X}) \cdot u_j^i = \sum_i p_i(\mathbf{X}) \cdot \sum_j u_j^i = \sum_i p_i(\mathbf{X}) \cdot F(\mathbf{X}) = F(\mathbf{X}).$$

In equilibrium, agent j accepts a proposal if its expected cost is smaller than the expected cost from rejecting it. As strategies are stationary and negotiations break down with probability $1 - \delta$, the expected cost of rejection is $\delta\mathbb{E}\pi_j + (1-\delta)f_j^0$, and the agent rejects all more expensive proposal. Thus, this is the equilibrium acceptance threshold: $a_j = \delta\mathbb{E}\pi_j + (1-\delta)f_j^0$. As $\sum_j f_j^0 = F(\mathbf{D})$ and $\sum_j \mathbb{E}\pi_j = F(\mathbf{X})$ as just shown,

$$\textstyle\sum_j a_j = \delta \sum_j \mathbb{E}\pi_j + (1-\delta)\sum_j f_j^0 = \delta F(\mathbf{X}) + (1-\delta)F(\mathbf{D}).$$

In equilibrium, agents make proposals to minimize their own cost by offering the others exactly their acceptance thresholds. That is, agent i assigns cost $u_j^i = a_j$ to agent j and $u_i^i = F(\mathbf{X}) - \sum_{j \ne i} a_j$ to themself.

Therefore,

$$
\begin{aligned}
\mathbb{E}\pi_j = \textstyle\sum_i p_i(\mathbf{X}) \cdot u_j^i &= p_j(\mathbf{X}) \cdot (F(\mathbf{X}) - \textstyle\sum_{i \neq j} a_i) + (1 - p_j(\mathbf{X})) \cdot a_j \\
&= a_j - p_j(\mathbf{X}) \cdot (F(\mathbf{X}) - \textstyle\sum_i a_i) \\
&= a_j - p_j(\mathbf{X}) \cdot (F(\mathbf{X}) - \delta \cdot F(\mathbf{X}) - (1-\delta) \cdot F(\mathbf{D})) \\
&= a_j - (1-\delta) \cdot p_j(\mathbf{X}) \cdot (F(\mathbf{X}) - F(\mathbf{D})) \\
&= a_j - (1-\delta) \cdot p_j(\mathbf{X}) \cdot \Delta(\mathbf{X}) \\
&= \delta \mathbb{E}\pi_j + (1-\delta) \cdot f_j^0 - (1-\delta) \cdot p_j(\mathbf{X}) \cdot \Delta(\mathbf{X}) \\
&= \delta \mathbb{E}\pi_j + (1-\delta) \cdot (f_j^0 - p_j(\mathbf{X}) \cdot \Delta(\mathbf{X})).
\end{aligned}
$$

Rearranging, $\mathbb{E}\pi_j = f_j^0 - p_j(\mathbf{X}) \cdot \Delta(\mathbf{X})$ as desired.

B.2 Proof of Proposition 2

('If' part) From Proposition 1, the expected cost for agent j is $f_j(\sum_t d_{jt}) - p_j(\mathbf{X}) \cdot \Delta(\mathbf{X})$ whenever $\Delta(\mathbf{X}) > 0$ and $f_j(\sum_t d_{jt})$ otherwise. Since, by assumption, j's recognition probability only depends on factors that j cannot affect, j can only influence her expected cost through $\Delta(\mathbf{X})$. If $p_j(\mathbf{X}) > 0$, maximizing Δ by choosing according to a cost-effective allocation is the best response, whereas if $p_j(\mathbf{X}) = 0$, it is one out of several best responses. Therefore, the best she can do is to choose x_j to maximize Δ, knowing that the agents that follow will in the same way choose announcements to minimize total costs. Since there is common knowledge, agents can compute a cost-effective allocation $\mathbf{X} \in \mathcal{E}(\mathbf{D}, \mathbf{C}, f)$ and choose their announcements accordingly, and this will be an equilibrium.

('Only if' part) By contradiction, suppose that there is $\mathbf{D} \in \tilde{\mathbb{D}}$, $\mathbf{C} \geq \mathbf{D}$ and $\mathbf{X}, \mathbf{Y} \in \mathcal{X}(\mathbf{D}, \mathbf{C})$ that coincide prior to agent j, $\mathbf{X}_{<j} = \mathbf{Y}_{<j}$, yet $p_j(\mathbf{X}) \neq p_j(\mathbf{Y})$. We will show that, when that is the case, there are cost functions for which j's expected cost is lower from announcing an inefficient workload compared to a cost-effective one, a contradiction to *cost-effective implementation*. We proceed from the end, starting with $j = n$, then $j = n-1$, and so on.

(Agent n) As agent n must announce x_n such that $\sum_i x_i = \sum_i d_i$, agent n does not have a different announcement $y_n \neq x_n$ to make. Hence, there are no such two allocations $\mathbf{X}$ and $\mathbf{Y}$, so $j \neq n$.

(Agent $n-1$) Next, consider agent $n-1$ who chooses x_{n-1} and implicitly x_n. Let $\mathbf{X}, \mathbf{Y} \in \mathcal{X}(\mathbf{D}, \mathbf{C})$ be such that $\mathbf{X}_{<n-1} = \mathbf{Y}_{<n-1}$ and $\mathbf{X} \neq \mathbf{Y}$. Suppose $p_{n-1}(\mathbf{X}) \neq p_{n-1}(\mathbf{Y})$ and, without loss, specifically that $p_{n-1}(\mathbf{X}) < p_{n-1}(\mathbf{Y})$. As p is continuous in the announcements, we may assume that $\mathbf{X}, \mathbf{Y} \neq \mathbf{D}$[11] and that, for each agent $k \geq n-1$, $\sum_t x_{kt} \neq \sum_t y_{kt}$. We proceed to show that there are costs f for which $\mathbf{X}$ is cost effective but agent $n-1$ is better off at the inefficient $\mathbf{Y}$.

Define $\ell_j(z) = \max\{z - \max\{\sum_t x_{jt}, \sum_t y_{jt}\}, 0\}$ and restrict attention to costs f such that, for each agent j and amount $z \geq 0$, $\ell_j(z) \leq f_j(z) \leq l_j(z) + \varepsilon$; see Figure 3. There exist such functions f for which marginal costs are equal across agents at $\mathbf{X}$, $f_i'(\sum_t x_{it}) = f_j'(\sum_t x_{jt})$, so $\mathbf{X} \in \mathcal{E}(\mathbf{D}, \mathbf{C}, f)$ and $\mathbf{Y} \notin \mathcal{E}(\mathbf{D}, \mathbf{C}, f)$.[12] By construction, for each agent i,

$$
f_i(\textstyle\sum_t x_{it}) \geq \ell_i(\sum_t x_{it}) = \ell_i(\sum_t y_{it}) \geq f_i(\sum_t y_{it}) - \varepsilon \implies \sum_i f_i(\sum_t x_{it}) \geq \sum_i (f_i(\sum_t y_{it}) - \varepsilon).
$$

[11] If, say $\mathbf{X} = \mathbf{D}$, then there is $\mathbf{X}' \neq \mathbf{D}$ close to $\mathbf{X}$ such that $p_{n-1}(\mathbf{X}') \approx p_{n-1}(\mathbf{X}) < p_{n-1}(\mathbf{Y})$, so we can use $\mathbf{X} = \mathbf{X}'$ instead.

[12] In particular, any marginal cost $\delta \in (0,1)$ with $\delta \leq \varepsilon / (\max_j \sum_t y_{jt} - \sum_t x_{jt})$ should be attainable.

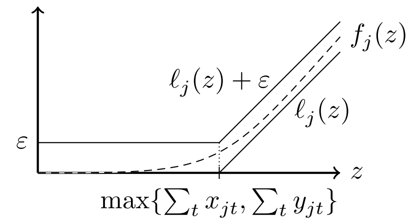

Figure 3: The cost function f_j (dashed) is within the bounds set by the piece-wise linear functions.

Hence, $F(\mathbf{X}) \geq F(\mathbf{Y}) - n\varepsilon$ and $\Delta(\mathbf{X}) = F(\mathbf{D}) - F(\mathbf{X}) \leq F(\mathbf{D}) - F(\mathbf{Y}) + n\varepsilon = \Delta(\mathbf{Y}) + n\varepsilon$. Since $\mathbf{X} \in \mathcal{E}(\mathbf{D}, \mathbf{C}, f)$ and $\mathbf{X} \neq \mathbf{D}$, $\Delta(\mathbf{X}) > 0$, so for small enough $\varepsilon > 0$, also $\Delta(\mathbf{Y}) > 0$. Therefore, since $p_{n-1}(\mathbf{X}) < p_{n-1}(\mathbf{Y})$, there exists $\varepsilon > 0$ such that

$$\frac{\Delta(\mathbf{X})}{\Delta(\mathbf{Y})} \leq \frac{\Delta(\mathbf{Y}) + n\varepsilon}{\Delta(\mathbf{Y})} < \frac{p_{n-1}(\mathbf{Y})}{p_{n-1}(\mathbf{X})}.$$

But then $n-1$ prefers the inefficient $\mathbf{Y}$ over the cost effective $\mathbf{X}$, a contradiction. Hence, $p_{n-1}(\mathbf{X}) = p_{n-1}(\mathbf{Y})$. As in the first part of the proof, agent $n-1$ therefore always announces workloads to maximize Δ.

(Agent $j < n-1$) Next, consider any agent $j < n-1$. By the argument above, j chooses announcement foreseeing that each agent $k > j$ will choose to minimize total costs given the choices made by agents prior to k. We proceed as for agent $n-1$.

Let $\mathbf{X}, \mathbf{Y} \in \mathcal{X}(\mathbf{D}, \mathbf{C})$ be such that $\mathbf{X}_{<j} = \mathbf{Y}_{<j}$ and $\mathbf{X} \neq \mathbf{Y}$. Suppose $p_j(\mathbf{X}) \neq p_j(\mathbf{Y})$ and, without loss, specifically that $p_j(\mathbf{X}) < p_j(\mathbf{Y})$. As p is continuous in the announcements, we may assume that $\mathbf{X}, \mathbf{Y} \neq \mathbf{D}$ and that, for each agent $k \geq j$, $\sum_t x_{kt} \neq \sum_t y_{kt}$. We proceed to show that there are costs f for which $\mathbf{X}$ is cost effective but agent j is better off at the inefficient $\mathbf{Y}$. Again, we bound the costs f by ℓ and $\ell + \varepsilon$ as when considering agent $n-1$ to ensure that $\mathbf{X}$ is cost effective, yet the total costs at $\mathbf{Y}$ are close to those at $\mathbf{X}$. When agent j changes announcement from x_j to y_j, then the agents who follow will still seek to minimize costs thereafter. That is to say, the costs f should also be such that $\mathbf{Y}_{>j}$ minimize total costs given $\mathbf{Y}_{\leq j}$.

By assumption, $\sum_t x_{jt} \neq \sum_t y_{jt}$. Consider first $\sum_t x_{jt} < \sum_t y_{jt}$. Suppose, for contradiction, that $\sum_t x_{kt} < \sum_t y_{kt}$ for some agent $k > j$. As there are fewer tasks left to complete for agents $j+1, \ldots, n$ following $\mathbf{Y}_{\leq j}$ than $\mathbf{X}_{\leq j}$, there is also an agent $k' > j$ for which $\sum_t x_{kt} > \sum_t y_{kt}$. However, by construction of f, marginal costs are equal at $\mathbf{X}$ for agent $j+1, \ldots, n$. As cost functions are strictly convex, the marginal cost must then be higher for agent k than for agent k' at $\mathbf{Y}$, a contradiction to that k and k' minimize total costs given the prior announcements. Hence, we have $\sum_t x_{kt} > \sum_t y_{kt}$ for each agent $k > j$. Therefore, we can select f such that marginal costs are equal at $\sum_t y_{kt}$ for all agents $k > j$ while simultaneously equating marginal costs for all agents at $\mathbf{X}$; see Figure 4. The second case, $\sum_t x_{jt} > \sum_t y_{jt}$, is analogous but instead results in $\sum_t x_{kt} < \sum_t y_{kt}$ for each agent $k > j$.

The proof proceeds by the same steps as for agent $n-1$: f is selected such that $\Delta(\mathbf{X}) \leq \Delta(\mathbf{Y}) - n\varepsilon$, so there exists a small enough $\varepsilon > 0$ for which agent j prefers the inefficient allocation $\mathbf{Y}$ over the cost-effective allocation $\mathbf{X}$, a contradiction to *cost-effective implementation*.

B.3 Proof of Proposition 3

('If' part) Consider the recognition function p associated with matrices $\mathbf{Z}^1, \mathbf{Z}^2, \ldots$ and an arbitrary problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$. It is immediate that p satisfies *top-agent proportionality*. For *consistency*, we will show

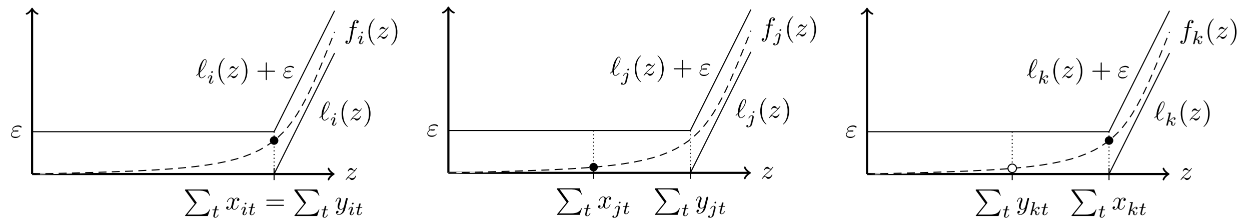

Figure 4: Sketch for $\sum_t x_{jt} < \sum_t y_{jt}$. Left is for agents $i < j$, middle for agent j, and right for agents $k > j$. Marginal costs should be equal across agents at the filled dots and likewise at the non-filled dots (for $k > j$).

that p is consistent with respect to $\mathbf{Z}^1$. Let $g(1), \dots, g(n)$ denote the resulting functions when applying p on $(\mathbf{X}, \mathbf{D}, \mathbf{C})$ and let $\tilde{g}(2), \dots, \tilde{g}(n)$ denote the corresponding functions when applying p on the reduced problem $(\mathbf{X}_{\geq 2}, \mathbf{D}_{\geq 2} + \mathbf{Z}^1_{\geq 2}, \mathbf{C}_{\geq 2})$.

By construction, we have $\sum_{i \geq 2} z_i^1 = e_1 - x_1$. For each $j \geq 2$,

$$\tilde{g}(j) = \frac{(\sum_t d_{jt} + \sum_t z^1_{jt}) + \sum_{2 \leq i < j} \sum_t z^i_{jt}}{(\sum_{i \geq 2} \sum_t (d_{it} + z^1_{it})) - \sum_{2 \leq i < j} \sum_t x_{it}} = \frac{\sum_t d_{jt} + \sum_{i<j} \sum_t z^i_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j} \sum_t x_{it}} = g(j).$$

Therefore, for each $j \geq 2$,

$$\begin{aligned} p_j(\mathbf{X}_{\geq 2}, \mathbf{D}_{\geq 2} + \mathbf{Z}^1_{\geq 2}, \mathbf{C}_{\geq 2}) \cdot (1 - p_1(\mathbf{X}, \mathbf{D}, \mathbf{C}) &= (1 - \tilde{g}(2)) \cdots (1 - \tilde{g}(j-1)) \cdot \tilde{g}(j) \cdot (1 - g(1)) \\ &= (1 - g(2)) \cdots (1 - g(j-1)) \cdot g(j) \cdot (1 - g(1)) \\ &= p_j(\mathbf{X}, \mathbf{D}, \mathbf{C}), \end{aligned}$$

showing that p is consistent with respect to $\mathbf{Z}^1$.

('Only if' part) Let p satisfy *consistency* and *top-agent proportionality* and fix an arbitrary problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$. By *consistency*, p is consistent with respect to some $\mathbf{Z}^1, \dots, \mathbf{Z}^{n-1}$ such that $\mathbf{Z}^1 \in \mathbb{Z}(\mathbf{X}, \mathbf{D}, \mathbf{C})$ and, for each $j \geq 2$, $\mathbf{Z}^j \in \mathbb{Z}(\mathbf{X}_{\geq j}, \mathbf{D}_{\geq j} + \mathbf{Z}^1_{\geq j} + \dots + \mathbf{Z}^{j-1}_{\geq j}, \mathbf{C}_{\geq j})$. We will show that for each $j \geq 2$,

$$p_j(\mathbf{X}, \mathbf{D}, \mathbf{C}) = (1 - g(1)) \cdots (1 - g(j-1)) \cdot g(j),$$

where g is as in the statement of Proposition 3 when applied to $\mathbf{Z}^1, \dots, \mathbf{Z}^{n-1}$. By *top-agent proportionality*,

$$p_1(\mathbf{X}, \mathbf{D}, \mathbf{C}) = \frac{\sum_t d_{1t}}{\sum_i \sum_t d_{it}} = g(1)$$

and, for each $j \geq 2$,

$$p_j(\mathbf{X}_{\geq j}, \mathbf{D}_{\geq j} + \mathbf{Z}^1_{\geq j} + \dots + \mathbf{Z}^{j-1}_{\geq j}, \mathbf{C}_{\geq j}) = \frac{\sum_t d_{jt} + \sum_{i<j} \sum_t z^i_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j} \sum_t x_{it}} = g(j).$$

Finally, by repeated application of *consistency*, for each $j \geq 2$,

$$\begin{aligned} p_j(\mathbf{X}, \mathbf{D}, \mathbf{C}) &= (1 - p_1(\mathbf{X}, \mathbf{D}, \mathbf{C})) \cdot p_j(\mathbf{X}_{\geq 2}, \mathbf{D}_{\geq 2} + \mathbf{Z}^1_{\geq 2}, \mathbf{C}_{\geq 2}) \\ &= (1 - p_1(\mathbf{X}, \mathbf{D}, \mathbf{C})) \cdot (1 - p_2(\mathbf{X}_{\geq 2}, \mathbf{D}_{\geq 2} + \mathbf{Z}^1_{\geq 2}, \mathbf{C}_{\geq 2})) \cdot p_j(\mathbf{X}_{\geq 3}, \mathbf{D}_{\geq 3} + \mathbf{Z}^1_{\geq 3} + \mathbf{Z}^2_{\geq 3}, \mathbf{C}_{\geq 3}) \\ &\vdots \\ &= \prod_{i<j} (1 - p_i(\mathbf{X}_{\geq i}, \mathbf{D}_{\geq i} + \mathbf{Z}^1_{\geq i} + \dots + \mathbf{Z}^{i-1}_{\geq i}, \mathbf{C}_{\geq i}) \cdot p_j(\mathbf{X}_{\geq j}, \mathbf{D}_{\geq j} + \mathbf{Z}^1_{\geq j} + \dots + \mathbf{Z}^{j-1}_{\geq j}, \mathbf{C}_{\geq j}) \\ &= (1 - g(1)) \cdots (1 - g(j-1)) \cdot g(j). \end{aligned}$$

B.4 Proof of Proposition 4

(Functional form) Take as given a problem $(\mathbf{X}, \mathbf{D}, \mathbf{C}) \in \mathcal{P}^n$. By Proposition 3, *consistency* and *top-agent proportionality* together imply that each probability $p_j(\mathbf{X}, \mathbf{D}, \mathbf{C})$ takes on a particular form, namely

$$p_j(\mathbf{X}, \mathbf{D}, \mathbf{C}) = (1 - g(1)) \cdot (1 - g(2)) \cdots (1 - g(j-1)) \cdot g(j),$$

where g is constructed with respect to some matrices $\mathbf{Z}^1, \ldots, \mathbf{Z}^{n-1}$ such that, in general,

$$g(j) = \frac{\sum_t d_{jt} + \sum_{i<j} \sum_t z^i_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j} \sum_t x_{it}}.$$

(Implementation) By Proposition 2, p satisfies *cost-effective implementation* if and only if, for each $i \leq j$, agent i's probability p_i is independent of agent j's announcement x_j. This is immediate for $p_1(\mathbf{X}, \mathbf{D}, \mathbf{C}) = g(1)$, as $g(1)$ is independent of $\mathbf{X}$. Assume now that for each $i < j$, p_i is independent of $\mathbf{X}_{\geq i}$. Then, for each $i < j$, $g(i)$ is independent of $\mathbf{X}_{\geq i}$. Therefore, p_j is independent of $\mathbf{X}_{\geq j}$ if and only if $g(j)$ is. Within $g(j)$, only the numerator $d_j + \sum_{i<j} z^i_j$ can depend on x_j.

(Maximal announcement) Given the announcements $\mathbf{X}_{<j}$ by agents prior to agent j, the maximal announcement that j can make is $\bar{x}_j = \min\{\sum_{i \leq j} d_i - \sum_{i<j} x_i, c_j\}$. For any announcement x_j, $\mathbf{Z}^1$ through $\mathbf{Z}^{j-1}$ will be such that $x_j \leq d_j + \sum_{i<j} z^i_j \leq \bar{x}_j$: the left inequality as otherwise j is unable to complete the announced tasks, the right due to the reasons just given. In particular, if j announces $\bar{x}_j$, then we have $d_j + \sum_{i<j} z^i_j = \bar{x}_j$. On the other hand, were j to announce some $x_j \leq \bar{x}_j$, then *cost-effective implementation* through Proposition 2 requires j's recognition probability to be unchanged. (Else there exists a problem in which x_j is cost-effective but $\bar{x}_j$ preferred to j or vice versa.) Specifically, also $g(j)$ and especially $d_j + \sum_{i<j} z^i_j$ needs to be unchanged. Hence, $\mathbf{Z}^1$ through $\mathbf{Z}^{j-1}$ must be such that $d_j + \sum_{i<j} z^i_j = \bar{x}_j$ regardless of j's announcement x_j. We will show that this is true if and only if, for each $i < j$, $\mathbf{Z}^i \in \mathbb{Z}^*(\mathbf{X}_{\geq i}, \mathbf{D}_{\geq i} + \mathbf{Z}^1_{\geq i} + \cdots + \mathbf{Z}^{i-1}_{\geq i}, \mathbf{C}_{\geq i})$.

($\mathbb{Z}^*$ necessary) Assume that, for some i, $\mathbf{Z}^i \notin \mathbb{Z}^*(\mathbf{X}_{\geq i}, \mathbf{D}_{\geq i} + \mathbf{Z}^1_{\geq i} + \cdots + \mathbf{Z}^{i-1}_{\geq i}, \mathbf{C}_{\geq i})$. For convenience and without loss, say $i = 1$. There is a first agent $j > i$ for which z^i_j is not as prescribed for $\mathbb{Z}^*$, so too much of the excess is pushed beyond j, and too little is left to j. (The proof is considerably easier for $j = 2$ than for $j > 2$. For that reason, we restrict attention to $j > 2$.)

Once all agents $i < j$ have been removed, agent j holds the following number of tasks:

$$d_j + \sum_{i<j} z^i_j = d_j + z^1_j + \sum_{1<i<j} z^i_j \leq d_j + z^1_j + \sum_{1<i<j} (d_i + z^1_i - x_i) = \sum_{1<i\leq j} (d_i + z^1_i) - \sum_{1<i<j} x_i.$$

The weak inequality follows as j receives at most $d_i + z^1_i - x_i$ out of the $d_i + z^1_i$ tasks that agent $i < j$ holds after completing their announced x_i tasks following 1's removal.

Suppose first that agent j falls into Case 1, so $\sum_{i \leq j} d_i \leq \sum_{i<j} x_i + c_j$ and $\bar{x}_j = \sum_{i \leq j} d_i - \sum_{i<j} x_i$, yet $\mathbf{Z}^1$ is such that $\sum_{i \leq j} z^1_i < d_j - x_1$. As $z^1_1 = 0$,

$$\sum_{1<i\leq j} (d_i + z^1_i) - \sum_{1<i<j} x_i < \sum_{i \leq j} d_i - \sum_{i<j} x_i = \bar{x}_j.$$

Suppose instead that j falls into Case 2, so $\sum_{i \leq j} d_i > \sum_{i<j} x_i + c_j$ and $\bar{x}_j = c_j$, yet $\mathbf{Z}^1$ contradicts equation $\star$ through $\sum_{1<i\leq j} (d_i + z^1_i) < \sum_{1<i<j} x_i + c_j$. Then

$$\sum_{1<i\leq j} (d_i + z^1_i) - \sum_{1<i<j} x_i < c_j = \bar{x}_j.$$

In both cases, had j instead announced $x_j = \bar{x}_j$, then $d_j + \sum_{i<j} z_j^i = \bar{x}_j$. Therefore, agent j can affect their probability p_j through their announcement x_j, a contradiction to *cost-effective implementation*.

($\mathbb{Z}^*$ sufficient) Fix an announcement x_j. Assume, for each $i < j$, $\mathbf{Z}^i \in \mathbb{Z}^*(\mathbf{X}_{\geq i}, \mathbf{D}_{\geq i} + \mathbf{Z}^1_{\geq i} + \cdots + \mathbf{Z}^{i-1}_{\geq i}, \mathbf{C}_{\geq i})$. We will show that $d_j + \sum_{i<j} z_j^i = \bar{x}_j$. First, suppose that, for each $i < j$, all excess is shared among agents $i+1, \ldots, j$. That is, for each $i < j < k$, $z_k^i = \vec{0}$. By construction, once all agents $i < j$ are removed, agent j holds as much as possible up to capacity, so $d_j + \sum_{i<j} z_j^i = \min\{\sum_{i \leq j} d_i - \sum_{i<j} x_i, c_j\} = \bar{x}_j$.

Suppose instead that there is a type t and agents $i < j < k$ such that $z_{kt}^i > 0$.[13] Identify step $i^* < j$ as the last step for which this holds, so, for each $i^* < i < j < k$, $z_k^i = \vec{0}$. By equation $\star$ applied on the reduced problem $(\mathbf{X}_{\geq i^*}, \mathbf{D}_{\geq i^*} + \mathbf{Z}^1_{\geq i^*} + \cdots + \mathbf{Z}^{i^*-1}_{\geq i^*}, \mathbf{C}_{\geq i^*})$, we set $z_j^{i^*}$ such that

$$\sum_{i^*<i\leq j}((d_i + z_i^1 + \cdots + z_i^{i^*-1}) + z_i^{i^*}) \geq \sum_{i^*<i<j} x_i + c_j.$$

Rearranging, we have

$$d_j + \sum_{i\leq i^*} z_j^i + \sum_{i^*<i<j}(d_i + z_i^1 + \cdots + z_i^{i^*} - x_i) \geq c_j \geq \bar{x}_j.$$

After step i^*, every intermediate agent i, so $i^* < i < j$, will complete x_i out of the $d_i + z_i^1 + \cdots + z_i^{i^*}$ tasks they hold after agents 1 through i^* have been removed. By the choice of i^*, no excess is pushed beyond agent j for $i^* < i < j$. Hence, once all agents $i < j$ have been removed, the remaining excess is eventually added to agent j through $\mathbf{Z}^{i^*+1}$ through $\mathbf{Z}^{j-1}$:

$$\sum_{i^*<i<j} z_j^i = \sum_{i^*<i<j}(d_i + z_i^1 + \cdots + z_i^{i^*} - x_i).$$

Taken together,

$$d_j + \sum_{i<j} z_j^i = d_j + \sum_{i\leq i^*} z_j^i + \sum_{i^*<i<j} z_j^i = d_j + \sum_{i\leq i^*} z_j^i + \sum_{i^*<i<j}(d_i + z_i^1 + \cdots + z_i^{i^*} - x_i) \geq \bar{x}_j.$$

Feasibility constraints dictate that $d_j + \sum_{i<j} z_j^i \leq \bar{x}_j$, so $d_j + \sum_{i<j} z_j^i = \bar{x}_j$ as desired.

B.5 Proof of Theorem 1

For each agent j, p_j^* is independent of $X_{\geq j}$, so by Proposition 2, p^* satisfies *cost-effective implementation*. As p^* belongs to the class characterized in Proposition 3, p^* satisfies *consistency* and *top-agent proportionality*.

The converse follows readily from the proof of Proposition 4. For each agent j, regardless of j's announcement x_j, we must have

$$d_j + \sum_{i<j} z_j^i = \bar{x}_j = \min\{\sum_{i\leq j} d_i - \sum_{i<j} x_i, c_j\}.$$

This pins down the desired $g(j)$, namely as in the construction of p^*:

$$g(j) = \frac{\sum_t \bar{x}_{jt}}{\sum_i \sum_t d_{it} - \sum_{i<j}\sum_t x_{it}} = \frac{\sum_t \min\{\sum_{i\leq j} d_i - \sum_{i<j} x_i, c_j\}_t}{\sum_i \sum_t d_{it} - \sum_{i<j}\sum_t x_{it}}.$$

[13] For convenience, we drop the subscript t in the remainder of the proof.