The Economics of Crowd Organizations*†

Daron Acemoglu‡ Mohamed Mostagir§ Asuman Ozdaglar¶

Abstract

How should we assign tasks of unknown difficulties to workers of unknown skills? This is the central problem addressed in this paper and one that is common in many decentralized labor markets and crowdsourcing platforms. In these markets, employers may lack the requisite information about the difficulties of the tasks they want completed, and workers have private information about their skills as well as their own incentive constraints. These aspects combine to make the problem of finding and implementing the optimal allocation of workers to tasks difficult. We address this problem through crowd organizations: a crowd organization emerges endogenously as a response to incentives in the market place, and consists of crowd workers acting *as if* they belong to the same organization in a way that mitigates the informational difficulties described above. We show that the optimal structure of this organization is a skill hierarchy and that, despite the informational constraints, this organization can be implemented in a decentralized fashion through a simple pricing mechanism. Platforms can embed this mechanism into their existing practices as a tool for discovering the difficulties and prices of novel tasks and also as a way to classify workers according to their skills.

1 Introduction

Decentralized labor markets have become increasingly pervasive, with employers sourcing out many of their tasks to the crowd through platforms like Upwork, Freelancer, Amazon Mechanical Turk, and several others. Upwork alone processes three million jobs annually, with a pool of five million

*An earlier version circulated under the title "Managing Innovation in a Crowd".

†We thank Glenn Ellison, Luis Garicano, Karim Lakhani, Jonathan Levin, David Miller, and seminar participants at Duke, Johns Hopkins, Michigan, Microsoft Redmond, MIT, University of Texas-Austin, and University of Washington for useful comments and discussion. We gratefully acknowledge financial support from Draper Labs and the Toulouse Network for Information Technology (supported by Microsoft).

‡Department of Economics, MIT.

§ Ross School of Business, University of Michigan.

¶ Laboratory for Information and Decision Systems, MIT.

employers, twelve million crowd workers, and upwards of a billion dollars worth of work being done through the platform.[1] Such large numbers naturally lead to a sizable variation in the nature and difficulties of the tasks being processed as well as in the available skills in the workers pool, and a fundamental challenge is how to match a task to the right worker for it. This is particularly important because while there is an abundance of workers, there is a scarcity of high skills, and matching a high-skilled worker to an easy task can be wasteful. This problem is further exacerbated by the fact that, unlike organizations, crowdsourcing takes place in a decentralized virtual market where platforms have noisy information about the workers' skills and cannot unilaterally assign specific workers to specific tasks. Instead, workers are free to choose which tasks to work on, and hence the success and efficiency of these platforms hinge on addressing the voluntary participation and incentive constraints of the workers.

In contrast, the theory of organization capital posits that a fundamental strength of organizations is the access that they have to personnel information —in particular, information about the skills of their workers— and their ability to act on this information in a centralized fashion that avoids most of the inefficiencies that happen in a decentralized marketplace. As Prescott and Visscher (1980) write:

> The work force is not homogeneous: workers have different sets of skills and talents. Some tasks within the firm are performed better with workers of a particular aptitude, and the efficiency of the organization depends on how well individuals are matched to tasks at which they have a comparative advantage.

Decentralized markets, on the other hand, address these issues in different ways, with some platforms completely overhauling and redesigning their own approach over time. For example, TaskRabbit, a platform that initially operated through having workers bid how much they want to be paid for a task, completely revamped their matching system recently, getting rid of the bidding process altogether and instead utilizing a proprietary algorithm that presents an employer with a small set of workers to choose from. Platforms like LiveOps, which let crowd workers answer customer service calls on behalf of employers, operate through a contest-like mechanism where workers are ranked based on skill and past performance and are assigned to calls based on their rankings. Upwork and Amazon Mechanical Turk let employers post tasks as well as prices (i.e. how much they are willing to

[1] https://digit.hbs.org/submission/upwork-connecting-freelancers-to-gigs-on-demand/

pay) and let workers pick up the tasks they want to work on. Pricing individual tasks, however, is not a trivial exercise, and incorrect prices can lead to a variety of inefficiencies, from task starvation (see, e.g. Faridani et al. (2011)), where no one picks up a task because it is priced too low, to unnecessary capital expenditure that results from overpaying for tasks that are not too difficult.

The goal of this paper is to simultaneously address the issues of matching and pricing in these decentralized markets through *crowd organizations*. A crowd organization emerges endogenously as an equilibrium of the incentives offered in the marketplace, and consists of crowd workers behaving *as if* they are part of the same organization, thereby reaping the informational benefits mentioned above. An important distinction between traditional firms and the employers in these markets is that, unlike firms, employers may have incomplete information about the difficulties of their tasks. This can happen because of a variety of reasons. For example, employers may lack the requisite domain knowledge about the task they want completed (e.g. translation tasks, writing the terms and conditions clause for a mobile app, writing a script that extracts email addresses from a webpage, etc.), and evaluating difficulties can be an onerous, costly, and potentially subjective exercise. This aspect together with the noisy information about skills and the workers' constraints imply that there is incomplete information about both sides of the market, and the central question becomes how to assign tasks of unknown difficulties to workers of unknown skill.

Results The paper addresses the above problem by developing a market pricing mechanism that achieves the optimal allocation of workers to tasks and ensures that the pool of skills is efficiently utilized. The main conceptual contribution of the paper is to show that despite the informational and coordination difficulties described above, crowdsourcing can be managed as if it was taking place *within* an organization, but with some constraints resulting from the market allocation. In particular, we show that by providing appropriate rewards and incentives in the marketplace, the problem of task allocation can be mapped to the problem that an organization faces in assigning workers of different but known skills to tasks of unknown difficulties.

This insight is formalized through first characterizing the centralized solution where tasks have unknown difficulties but worker skills are known and workers can be assigned to sets of tasks (and thus their voluntary participation decisions can be ignored). Under these assumptions, the optimal organization takes the form of a *skill hierarchy*: workers are arranged into groups in ascending order

of skill and tasks are offered to groups in a sequential fashion. The structure of the optimal organization is derived via a dynamic programming formulation. The most distinctive property of that organization is that it ensures that the heterogeneous skills of the available workers are efficiently utilized.

We next show that this optimal organization can be implemented when worker skills are unknown and workers' participation (and in particular, their decision of *where* to participate) is voluntary. The implementation relies on a pricing scheme in which rewards for completing tasks increase the longer these tasks remain uncompleted (i.e., the more times they have been unsuccessfully attempted by other workers). What makes this pricing scheme support the optimal matching is that, as we show, there is a natural *comparative advantage* property: if a worker of a given skill level prefers to participate later rather than earlier, then higher-skilled workers would also choose to do so.[2] Thus the increasing prices of difficult (uncompleted) tasks and comparative advantage jointly provide the correct incentives for higher-skilled workers to participate in the upper echelons of the hierarchy, where they would normally be placed in the centralized solution, and also incentivize low-skilled workers to participate in the lower levels in order to increase their chances of getting a task that is not too difficult for them to finish. By appropriately setting the prices, workers can be induced to self-select into the desired hierarchy and endogenously form an organization that acts 'collaboratively', in the sense of filtering easy tasks in the lower levels and passing difficult tasks up the hierarchy when they cannot be solved at the current level.

This approach to task pricing and assignment has several appealing features. First, it provides a market mechanism that automates the process of pricing individual tasks, thereby avoiding the associated pricing pitfalls discussed earlier. Second, it avoids pricing problems that arise from informational asymmetries between workers and employers. This point is related to the literature on experts, e.g. Wolinsky (1993), where the primary difficulty is judging whether the value of the expert's service is inflated or not. This difficulty is usually mitigated through developing reputations. However, Kamenica and Gentzkow (2011) show that an expert can still reveal information in a manipulative way even when all the parties involved are rational. In addition, experimental work by Dulleck et al. (2011) suggests that workers have no incentive to build reputations as long as they can

[2]See Roy (1951); Sattinger (1975); Tinbergen (1974); Teulings (1995) on the role of comparative advantage in other problems of assignment of workers to tasks.

avoid liabilities. Enforcing liabilities in such decentralized environments – where workers come from different regions with different regulations– can be challenging, and presents an undue burden and overhead for employers and platforms.

It is important to point out that the approach presented in this paper is best viewed as complementary to existing practices. Crowdsourcing platforms still utilize a mixture of pricing, reputations, and tournament-like mechanisms to solve the matching problem and deliver the best solutions to their users, and the methodology developed in this paper can be used as a tool that endogenously generates information about the different variables of interest (like task difficulties and worker skills). For example, platforms like Upwork often suggest specific workers and corresponding prices to employers based on the attributes of their tasks, and our approach can be embedded into such a system to learn and classify task attributes and prices (and worker types) over time in order to facilitate such suggestions. Novel tasks can be processed through the hierarchy to discover the correct price and the type of worker who can finish these tasks, while tasks whose attributes are similar to past tasks can be directly routed to specific workers at predetermined prices. Similarly, one can combine the benefits of hierarchies (automated discovery of difficulties and prices, and efficient use of existing skills) with those of tournaments (explicit modeling of effort and competition). Indeed, TopCoder rated competitions are divided into two divisions, and competitors in the lower division cannot participate in the higher one unless they attain a certain score – a direct proxy for a skill hierarchy.

Finally, we show that while the optimal hierarchy can be difficult to derive, an organization with only two levels achieves a substantial fraction of the throughput of the optimal hierarchy, regardless of how many levels that hierarchy has. This partition of workers into two levels or classes is widely observed in many settings. In addition to the aforementioned TopCoder example, Amazon Mechanical Turk has two categories: workers, and master workers. Even some crowd platforms that are not pecuniary in nature, like the website fold.it which lets players fold proteins online, make use of two categories of players, with the 'beginner' category not permitted to attempt tasks that are rated as too difficult. Our results: a) provide a framework that utilizes self-selection to endogenously achieve this partition and identify who the skilled workers are, and b) suggest that this simple division into two classes is enough to generate a substantial fraction of what the optimal arrangement of workers can achieve.

Related Literature The potential for using crowds to improve the decisions and operations of firms has attracted a lot of recent attention. Araman and Caldentey (2016) explore the idea of using a crowd voting mechanism to gauge demand and interest in a product before committing to production, Babich et al. (2018) study how crowds can be used to generate startup funding and compare these mechanisms to traditional funding sources, Huang et al. (2014) use data from Dells IdeaStorm to describe crowds potentially influencing the designs and offerings of the firm, and Lobel et al. (2015) study how crowd referrals can be used as an effective marketing tool.

A recent and related concept to our work is that of a flash organization. The term originated in the computer science literature in a crowdsourcing context: Retelny et al. (2014) and Valentine et al. (2017) build a platform that assembles teams of crowd workers on the fly to work collaboratively on a project. Building these teams crucially relies on project managers hiring workers for specific tasks, which requires more information about the tasks to be completed as well as the workers to be hired. In contrast, the focus in our work is on how to provide incentives in information-scarce environments so that an organization arises endogenously without explicit recruiting. This framework also has a conceptual overlap with service supply chains, e.g. Allon and Federgruen (2005), where firms outsource tasks to other firms, who can in turn outsource these tasks and so on, similar to how workers in organizations can push tasks that they cannot finish to other workers with the requisite skills.

The work in this paper shares some attributes with skill-based routing in call centers. In this setting, customers have different needs and service operators have different skills, and various papers (e.g. Gans et al. (2003); Wallace and Whitt (2005); Armony and Ward (2013)) study joint routing and staffing decisions in a queuing context. Because of the centralized nature of these systems, these papers do not consider incentives and instead focus on the stability of the queuing systems and customer satisfaction, whereas the work in this paper examines the problem when there is uncertainty about both worker skills and task difficulties as well as incentive issues, with the focus being on the decentralized assignment problem rather than the queuing aspect.

Another relevant strand of literature is the work on performance-based rankings. These are mechanisms where several competitors attempt the same task and employers can choose and reward the best outcome(s), e.g. Moldovanu and Sela (2001); Terwiesch and Xu (2008). This setup is most useful for tasks where exposure to a wide pool of competitors might generate different solutions to the

same problem and where there might be a subjective measure to the outcome (e.g. logo design), with the emphasis being on the number of competitors to allow into the contest and optimal prize division. The focus of our paper is instead on one-to-one assignment models similar to those found at Upwork, Freelancer, or Amazon Mechanical Turk, where a task is assigned to a single worker. We abstract away from subjective or quality judgments by focusing on tasks that are binary in nature, so that a task can either be completed to specifications or not, e.g. writing a script that plots the spectral content of an audio signal or that flags online messages if they contain certain words.

This paper is also closely related to the assignment and assortative matching models in the economics literature, e.g. Koopmans and Beckmann (1957); Sattinger (1975), where heterogeneous skills and task difficulties lead to hierarchies through central or market allocations in a full information setting. The literature on knowledge hierarchies, e.g. Garicano (2000); Beggs (2001); Fuchs et al. (2015) brings this problem to settings where task difficulties are unknown and the decision of what skills to acquire is endogenous to the workers and the firm. This decision is based on the difficulty distribution of the tasks that the firm faces (for example, a firm can hire or train its workers to deal with a specific range of tasks), and therefore ensures that the skills in the worker pool and the difficulties in the task pool are aligned. This feature, though it makes these models elegantly tractable, does not provide an adequate description of crowdsourcing environments, where skills are often dispersed and unknown, mismatches between task difficulties and skills are commonplace, and where voluntary participation of workers is a central issue. We contribute to that literature by incorporating these elements into the matching framework.

Finally, our model can be viewed as a specific instance of stochastic matching problems, where nodes in a graph are connected by edges and these edges have 'success probabilities' attached to them. A matching of two nodes is successful with a probability equal to the probability of the edge that joins these nodes, and success (or failure) is realized upon the actual match. In our setting, one node is a worker and the other is a task, and the probability of successful match indicates how likely it is that the difficulty of the task is within the worker's skill. There has been recent interest in this problem because of its applications to advertising, online dating, and kidney allocation problems, for example, as in Feldman et al. (2009); Manshadi et al. (2012). The stochastic nature of the problem leads to an exponentially large dynamic programming formulation with no adequate approximate solution

(see, e.g., Chen et al. (2009)). Our focus in this paper is not on computing the exact optimal matchings, but on understanding the structural properties of these matchings and their implications to how platforms and firms should organize their crowdsourcing efforts. Because the supply of workers in these environments is usually very large, we circumvent some of the technical problems encountered in the stochastic matching literature by studying the continuum limit of these markets. Ashlagi and Shi (2015) utilize a similar approach to characterize the optimal matchings in school choice.

The paper is organized as follows. Section 2 introduces the model and defines the centralized and implementation problems. Section 3 derives the necessary results to analyze the dynamic programming formulation of the centralized problem. Section 4 provides a pricing scheme for the implementation problem, and Section 5 discusses the role of the assumptions in our model, offers extensions, and concludes the paper.

2 Model

Setup. Our benchmark model considers *overlapping generations* of tasks. Each time period $\tau = 1, 2, ...$ a set of tasks T^τ arrives, and is processed by a set of workers W. The difficulty d of a task in T^τ is a nonnegative scalar that is drawn from a distribution F_D for all τ. Similarly, the skill s of a worker in W is a nonnegative scalar distributed according to F_S. The set W is indexed by $i \in [0, A]$, and we assume that workers are ordered in ascending order of skill, so that $s_i \geq s_j$ if $i > j$, where s_i and s_j are the skills of workers i and j, respectively. The sets W and T^τ are equipped with a Lebesgue measure λ, where $\lambda(T^\tau)$ is normalized to one for all τ and $\lambda(W) = A > \lambda(T^\tau) = 1$, i.e. there are more workers than there are tasks in any single group of tasks T^τ.

Inspired by the crowdsourcing environment where workers select which tasks to work on, we consider matchings that are decentralized and, as mentioned in the introduction, require only that these matchings are one-to-one, i.e. a task is matched to a worker if and only if that worker is matched to that task. Formally, in our continuum setting a matching $m : W \to T$ is a one-to-one measurable map satisfying measure consistency: for every open interval $Z \subseteq W$, we have $\lambda(Z) = \lambda(m(Z))$. We write $m(X, Y)$ to denote that the set of workers X and the set of tasks Y are matched. Define the output of a pairing of a worker of skill s with a task of difficulty d to be equal to 1 if $s \geq d$ and 0 otherwise, so that a worker successfully completes a task if the task difficulty is at most equal to the

worker's skill. We denote by $\theta(m)$ the output of matching m, which is the total measure of tasks successfully completed in m, i.e. the measure of task-worker pairs where the task difficulty is at most equal to the skill of the worker.

Each time period a worker can attempt a task and either completes it (if it is within his skill) or not. A task that is attempted by a worker but not completed can be attempted by another worker in the next time period. This allows the matching to be sequential. In period τ a set of tasks T^τ arrives, and is attempted by a set of workers in a first matching m_1^τ, then in period $\tau+1$ a second matching $m_2^{\tau+1}$ includes those tasks that were not completed in the first match and another set of workers and so on, thus the subscript i indicates that this matching includes tasks that have been unsuccessfully matched $i-1$ times before and the superscript is the time at which this matching is taking place. This means that there are multiple concurrent matches at any time $\tau > 1$: the tasks in the system are those that have just arrived, i.e. T^τ, and are matched to a set of workers, and the tasks that have not been completed in previous time periods and are matched to different sets of workers. Since a task in this framework can be matched several times, a sequential matching is likely to take longer to conclude than a one-shot matching. In addition, tasks that cannot be completed by anyone will stay in the system forever. We therefore assume that employers can afford to wait at most K time periods from the time of submission to the time tasks are returned (similar to the 'patience parameter' in the stochastic matching literature, e.g., Chen et al. (2009)). This means that tasks that enter the system at time τ exit at time at most $\tau+K$ regardless of whether they were completed or not. This idea is captured formally in the following definition and illustrated in Figure 1. We denote by W_i^τ and T_i^τ the sets of workers and tasks that constitute the match m_i^τ, and we let $\bar{T}_i^\tau$ denote the set of tasks *not* completed in m_i^τ.

Definition 1. (Sequential Matching) A sequential matching μ of tasks to workers is a sequence of matchings $\{m_i^\tau = m(W_i^\tau, T_i^\tau)\}$ that satisfy the following at time τ

1. $W_i^\tau \cap W_j^\tau = \emptyset$ for all τ, i, and $j \neq i$
2. $T_1^\tau = T^\tau$ for all τ and $T_i^\tau = \bar{T}_{i-1}^{\tau-1}$ for $2 \leq i \leq K$, $\tau \geq 2$.

The first part of the definition follows from the assumption that matchings are one-to-one: if at time τ a group of workers W_i^τ are matched to a group of tasks T_i^τ as part of matching m_i^τ then no

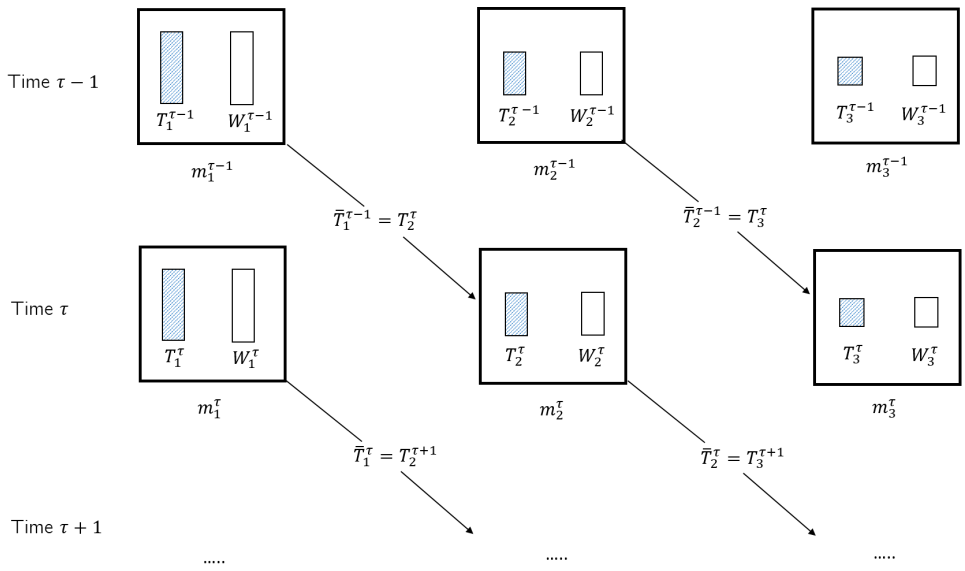

Figure 1: Sequential matching with $K = 3$ and a focus on time period τ. There are three concurrent matches happening in period τ: First, the new set of tasks arriving at τ go through their first matching as T_1^τ. Second, the tasks T_2^τ, which are the tasks that were not completed in their first match in the previous period $\tau - 1$, go through their second matching in period τ. Finally, tasks T_3^τ are the tasks going through their third and final matching after not being completed during periods $\tau-1$ and $\tau-2$. The process repeats in period $\tau + 1$: the tasks not completed in matching m_1^τ, denoted by $\bar{T}_1^\tau$, will go through their second matching in the next period $\tau + 1$ (where they are denoted as $T_2^{\tau+1}$). The tasks not completed in matching m_2^τ, denoted by $\bar{T}_2^\tau$, will go through their third matching in the next period $\tau + 1$ (where they are denoted as $T_3^{\tau+1}$). The tasks not completed in matching m_3^τ are discarded.

worker in that group can be part of any other matching that is taking place during the same time period (since a worker can only work on a single task in a time period). The second part states that the pool of tasks available for the i^{th} matching at time τ are those tasks that were not completed in the previous matching $m_{i-1}^{\tau-1}$.

We denote the throughput of the sequential matching μ at time τ by $\theta(\mu^\tau) = \sum_{i=1}^{K} \theta(m_i^\tau)$, so that the total throughput in a time period is the sum of completed tasks in the individual matchings happening during that period. We are interested in maximizing the *steady state* throughput, and so we drop the time superscript τ and focus on the time-invariant throughput $\theta(\mu)$. Note that for any $i > 1$, the tasks available for the i^{th} matching are *endogenously* determined based on the workers who were previously matched to these tasks. In other words, an allocation of workers to groups $W_1, ..., W_K$ determines the sets $T_2, ..., T_K$. In particular, we want to find an organization of workers into at most K groups such that the time-invariant sequential matching μ^* induced by the groups $W_1^*, ...W_K^*$ has

maximum throughput, i.e.

$$\max_{W_1,\ldots,W_K} \theta(\mu) \tag{1}$$

We consider the centralized and implementation versions of Problem 1:

The Centralized Problem The centralized problem involves solving (1) assuming that worker skills are known and workers can be allocated to different groups without respecting their incentive compatibility or participation constraints. The solution to this problem, even though it is only a step in the mathematical characterization of the desired market allocation and not our principal problem of interest, reveals some interesting structural properties that contrast with some of the results in the economics and organization design literatures. This is explored further in Section 3 and **??**.

The Implementation Problem The implementation problem characterizes the optimal organization subject to the additional informational and incentive compatibility constraints of workers — which are ignored in the centralized problem. This problem can in general be quite difficult in view of the myriad incentive compatibility constraints that may distort the optimal matching away from the "first-best" matching — the solution to the centralized problem. In our case, however, we are able to show that with an appropriate pricing scheme, the solution to the centralized problem can be implemented in a decentralized fashion. This eliminates the need for a characterization of "second-best" matching mechanisms, which is generally a much less tractable problem than the implementation of the first-best optimal mechanism.

3 The Centralized Problem

This section analyzes the optimal arrangement of workers in the absence of incentives and voluntary participation constraints. Several of the approximation algorithms for stochastic matching problems discussed in the introduction rely on greedy methods that in our context amount to assigning tasks to the most skilled workers first. The idea is that difficult tasks in the pool can impede the less skilled workers and compromise their productivity, and so having the more skilled workers clear out these tasks first will improve the output of the less skilled workers and the overall productivity of the system. As we discuss below, this is never a feature of the optimal organization: the opposite effect —that it is costly to expose skilled workers to easy tasks— is the main determinant of the

throughput of the system. The rest of this section formally proves this, derives the optimal structure, and concludes with some comparative statics, including the result that an organization with at most two levels achieves a substantial fraction of the throughput of the optimal organization, regardless of the number of levels that this organization has.

Before presenting the main structural result of this section, we give the following definition, which is a special case of Definition 1. Recall that our analysis is concerned with the time-invariant throughput, and hence the subscript τ is omitted from Definition 1 and all the following results and discussion.

Definition 2. (Hierarchical Matching) Consider a sequential matching and for all i, let $\min\{W_i\} = \{\min_j s_j | j \in W_i\}$ and $\max\{W_i\} = \{\max_j s_j | j \in W_i\}$. A sequential matching is **hierarchical** if $\max\{W_i\} \leq \min\{W_l\}$ whenever $i < l$.

A hierarchical matching is a sequential matching with the added property that workers are partitioned into groups based in ascending order of their skills, so that workers in group i are less skilled than workers in group $l > i$. Tasks are then offered to the least-skilled group, where some tasks may be completed and some not. The remaining tasks are assigned to the next group and the process repeats. With this definition in hand, the next proposition describes the structure of the optimal organization.

Proposition 1. *Consider a set T of tasks of unknown difficulties and a set of workers W with $\lambda(W) > \lambda(T)$; the throughput-maximizing sequential matching is a hierarchical matching.*

Proposition 1 states that the optimal organization of workers is a skill hierarchy, i.e. throughput is maximized by organizing workers in a hierarchical fashion according to skill and then assigning them tasks sequentially. The next series of results establish the structure and lead to a proof of the proposition, but we first illustrate the ideas through a simple example.

Example 1. Consider a set of tasks T with $\lambda(T) = 1$ and let the difficulty distribution of the tasks F_D be uniformly distributed over $(0, 1)$. Let W be a set of workers with $\lambda(W) = 2$ and a distribution of skills F_S, also uniformly distributed over $(0, 1)$. Let K, the number of groups in the organization, be equal to 1, then the optimal one-shot matching assigns all tasks to the most skilled group of workers whose size is equal to the size of the tasks. These are the unit measure of workers whose skills lie

in $(0.5, 1)$. The throughput of this matching is equal to 0.75. This is because any task with difficulty less than or equal to 0.5 will be finished with probability 1 (since even the least skilled worker in the group can do it) and a task with difficulty $0.5 < x < 1$ will be finished with probability $2(1-x)$ (for instance, since half the workers in the group have a skill that is equal to 0.75 or higher, a task of difficulty $x = 0.75$ has a chance equal to 0.5 of being finished), leading to a total throughput of

$$\frac{1}{2} \cdot 1 + \frac{1}{2} \int_{0.5}^{1} 2(1-x) f_D(x) dx = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}.$$

where $f_D(x) = 2$ for $x \in (0.5, 1]$.

Now let $K = 2$. The optimal grouping is as follows: W_1 contains all workers with skills in $(0.25, 0.75)$ and W_2 contains all workers whose skills are in $(0.75, 1)$. The matching m_1 has $\theta(m_1) = 0.5$ (since any task with difficulty less than 0.25 is finished with probability one, any task whose difficulty is over 0.75 is not finished, and any task whose difficulty is in $(0.25, 0.75)$ is finished with probability 0.5). The leftover tasks from this matching constitute the group T_2 which has measure 0.5 and difficulty distribution F_{D_2}. This distribution has two types of tasks, those that are in $(0.5, 0.75)$ and those that are in $(0.75, 1)$, with each type having an equal measure of 0.25. Matching $m_2 = m(W_2, T_2)$ assigns the remaining tasks to W_2 and has $\theta(m_2) = 0.375$, since all tasks with difficulties less than 0.75 will be finished (for a total measure of 0.25 completed), and tasks with difficulties in $(0.75, 1)$ will be finished with probability 0.5 (for a total measure of 0.125 completed). Thus the total output of this matching is equal to $\theta(m_1) + \theta(m_2) = 0.875$, which is significantly higher than the one shot matching output of 0.75.

Example 1 illustrates how an organization with two hierarchical groups was able to improve on the organization with a single group because in the latter, some of the skilled workers were assigned tasks that could have been done by other less skilled workers, resulting in under-utilization of their skills. The two-group solution filters out some or all of the easier tasks earlier in the process, so that there is a smaller chance that the skills of high-skilled workers in the second group are wasted on these tasks. In particular, as we ascened in the levels of an organization, tasks that would be too easy (meaning that they can be done by less skilled workers) are completed, leaving more difficult tasks to higher-skilled workers. It is worth noting that the optimal one-group solution uses only workers whose skills are above 0.5 and the two-group solution uses only workers whose skills are higher than

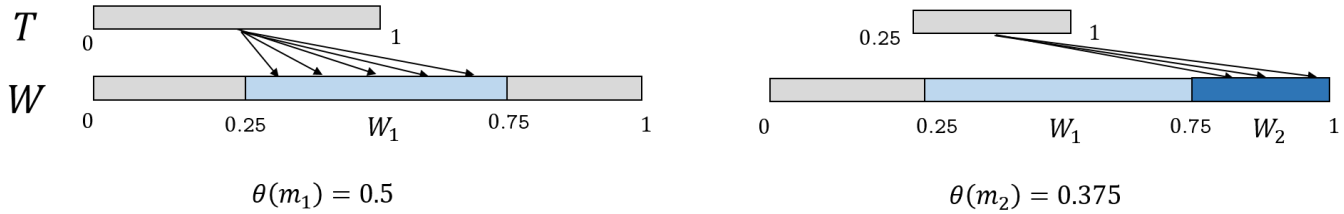

Figure 2: Sequential matching with $K = 2$ and twice as many workers as tasks. The task set T has unit measure and is first assigned to group W_1 of the same size and with workers whose skills are uniformly distributed over (0.25,0.75). Half of the tasks are finished in this matching and the other half T_2 has tasks that range in difficulty from 0.25 to 1. Task set T_2 is then assigned to group W_2 which has the same size and contains workers whose skills are uniformly distributed over (0.75,1). These workers are able to finish 0.375 out of the 0.5 tasks, making the total number of tasks completed in both matchings equal to 0.875. Workers whose skills are less than a threshold of 0.25 are not included in the matching.

0.25, i.e. it might sometimes be beneficial to restrict participation to only those workers whose skills are above a certain threshold. Finally, the optimal two-group solution is contiguous; once the first group contains workers of a certain skill, every worker whose skill is above that skill is also part of the organization. As we will see later, these properties hold for all optimal sequential matchings.

We now formalize the above intuition. Recall that for two random variables A and B we write $A \succ B$ to indicate that A first-order stochastically dominates B, i.e. $A \succ B$ implies that the distributions $F_A(.)$ and $F_B(.)$ satisfy $F_A(x) \leq F_B(x)$ for all x, with strict inequality for some x. The following two lemmas are central to the analysis of the characterization in Proposition 1.

Lemma 1. *(Stochastic Dominance) Let $m(W, T)$ be a matching and $\bar{T}$ be the tasks not completed in that matching. Let D_T and $D_{\bar{T}}$ be the random variables associated with task difficulties in T and $\bar{T}$, respectively, and assume that $\bar{T} \neq \emptyset$, then $D_{\bar{T}} \succ D_T$.*

Lemma 1 relates the pre- and post- matching difficulty distributions of tasks. It states that the difficulty distribution in $\bar{T}$ is skewed more towards difficult tasks than it is in T, and therefore the chances of selecting a difficult task from $\bar{T}$ is higher than it is in T. Note that if we were operating in a discrete domain, then it is indeed possible for any single instance that the difficulty distribution after a matching is easier than the distribution we started with. The possible permutations makes it difficult to derive the task distributions after successive matchings. Lemma 1 utilizes the continuum formulation to sidestep this difficulty.

We also define a notion of dominance for worker groups.

Definition 3. (Group dominance) A group of workers W_1 dominates group W_2 if $\min\{s_i : i \in W_1\} > \max\{s_j : j \in W_2\}$. We write this as $W_1 \succ W_2$.

Whether the binary relation $\succ$ is used to compare worker groups in the sense of Definition 3 or difficulty distributions in the first-order stochastic dominance sense will be clear from context. The next lemma examines what happens to the output of a matching when workers are rearranged between groups. Essentially, Lemma 2 is a specialization of Proposition 1 when $K = 2$.

Lemma 2. *(Swapping Lemma) Let W_1 and W_2 be two arbitrary groups where $W_2 \not\succ W_1$ and denote by μ^A the matching $\left(m_1^A, m_2^A\right) = \left(m(W_1^A, T_1^A), m(W_2^A, T_2^A)\right)$. There exists a rearrangement of workers in $W_1^A \cup W_2^A$ into two groups W_1^B and W_2^B such that $W_2^B \succ W_1^B$, $\mu^B = \left(m_1^B, m_2^B\right) = \left(m(W_1^B, T_1^B), m(W_2^B, T_2^B)\right)$, and $\theta(\mu^B) \geq \theta(\mu^A)$.*

The Swapping Lemma implies that the output of a matching that is obtained from two groups of workers of arbitrary composition can be improved upon by swapping some workers between the two groups to achieve a rearrangement where the workers in the second group all have higher skills than those in the first. It is important to point out that while the combined output of two groups increases when they are arranged in a hierarchical fashion, *the output of the dominant group can only decrease*, since by Lemma 1, the workers in that group now face a more difficult distribution than the one they would have faced if they came first in the ordering. In general, the outputs of the latter groups decrease as the skills in earlier groups increase. This is because more tasks are completed before they reach the latter group. This fact plays a particularly important role in the implementation problem because it implies that workers always prefer to be in an earlier group regardless of their skills, since these groups face easier difficulty distributions and therefore the chances of finishing a task there is higher. This indicates that successful implementation in the decentralized problem hinges on the ability to adjust workers' incentives in order to have them self-select into the appropriate group that they would be selected into in the centralized solution. In addition and as we saw in Example 1, the centralized solution might sometimes exclude all workers below a certain skill threshold, and replicating this solution again requires that these workers voluntarily choose not to participate in the decentralized solution.

We remark that having more groups does not necessarily increase throughput and can in fact decrease it. The intuition for two groups is the following: Consider a task that is too difficult for

almost all workers. In a one-shot setting, this task will lead to wasting the efforts of at most one worker, but as more groups are introduced, the same task has the potential to repeatedly 'defeat' other workers as well, as it gets carried through from one group to the next. These workers could have been better utilized in performing other tasks, which is what would happen in a one-shot setting. Thus breaking up a group into two may lead to an overall decrease in output.

Finally, we note that an organization with at most two groups is enough to generate a substantial fraction of the throughput of the optimal organization:

Proposition 2. *A hierarchical organization with at most two groups achieves at least 87.5% of the throughput of the optimal organization.*

Proposition 2 states that the throughput of the optimal organization can be approximated by an organization with only two levels, regardless of how many groups and workers are used in the optimal organization and regardless of what the optimal hierarchy looks like. This means that solving the dynamic program to find the optimal groups — which can be burdensome as the number of groups K becomes larger— is not necessary to construct organizations that achieve a substantial fraction of the optimal throughput. These two-tier arrangements are quite common in practice and can be seen, as discussed in the introduction, in several platforms such as TopCoder, Amazon Mechanical Turk, fold.it and so on.

4 The Implementation Problem

We return to the decentralized problem where worker skills are private information and workers themselves decide whether and when to participate. We show that with a simple pricing mechanism, we can nevertheless achieve the first-best allocation characterized in the previous section.

Formally, let the skill of worker w be equal to s and let the worker's participation cost be equal to q. The participation cost can be the time the worker takes to sign up for the platform, look for an appropriate task, and so on. Consider the case when there is no sequential matching. A worker who gets a task t successfully completes this task if it is within his skill s, and gets paid p. Fix the worker and task populations, and normalize the workers reservation wage to be equal to zero, then a worker participates if his expected payoff is nonnegative. Denote by $\psi(s, W, T)$ the probability that a worker

completes a task when his skill is s, the workers pool is W and the task pool is T, then a worker participates iff $p\psi(s, W, T) - q \geq 0$

When the matching is sequential, we denote by $\psi(s, W_i, T_i)$ the probability that a worker with skill s finishes a task in Group i, when the workers participating in that group and the tasks available are given by the sets W_i and T_i, respectively. Payment for successful completion of a task in Group i is equal to p_i. A worker then solves

$$\max_i p_i \psi(s, W_i, T_i) - q$$

Note that all workers concurrently solve the maximization problem and hence their participation decisions, the groups W_i (and consequently, the pool of tasks T_i available in matching i) are all endogenously determined.

The mechanism charges a participation or entry fee q for workers and a tiered payment system for completing tasks. A task finished in group $i, i = 1, ..., k$ pays an amount p_i, with $p_i > p_j$ for $i > j$. Fix a configuration of groups and note that the expected payment that a worker in in Group i receives is equal to $p_i\psi(s, W_i)$. Groups are in equilibrium when no worker can be strictly better off if they choose to move to another group.

Definition 4. (Group Equilibrium) Groups $W_1, ..., W_k$ are in equilibrium if a worker with skill s in $W_i, i = 1, ..., k$ receives expected payment $p_i\psi(s, W_i) \geq p_j\psi(s, W_j)$ for all $j \neq i$ and all s.

Thus for a worker of skill s to voluntarily choose to belong to group W_i, it must be that $p_i\psi(s, W_i) \geq p_j\psi(s, W_j)$ for all $j \neq i$. The following lemma is critical to the implementation of the optimal assignment.

Lemma 3. *(Comparative Advantage) Define $\phi(s) = \frac{\psi(s,W_i)}{\psi(s,W_j)}$ for $j > i$. Then $\phi(s)$ is decreasing in s.*

Comparative advantage follows from the monotone relationship between skills and difficulties. Lemma 3 states that higher-skilled workers perform *relatively* better on more difficult tasks than less skilled workers do. In the context of our model, this means that for two workers in group i with skills s_1 and s_2 and with $s_1 < s_2$, the decrease in success probability of the worker with skill s_1 upon moving to a higher group will be larger compared to the decrease for the worker with skill s_2 if he makes the same move. This implies that if the worker with skill s_1 prefers to be in a higher-indexed group, then so will a worker with skill s_2. The importance of this lemma is that it allows us to reduce

the incentive compatibility constraints to local constraints: instead of checking that the constraints hold for all workers, it is enough to only ensure that they hold for those workers at the boundaries.

We now give the main result of this section.

Proposition 3. *Let $W_1^*, ..., W_K^*$ be the centralized, optimal K-level hierarchy. There exists a participation fee q and payments $p_1 < p_2 < ...p_{K-1} < p_K$ that induce an equilibrium in which a worker chooses to be in group W_i^* if and only if the worker is selected for that group in the optimal allocation.*

Let $s_i = \{\min s | s \in W_i\}$. The proof of Proposition 3 shows that the values for the entry fee q and payments $p_1, ...p_K$ can be obtained by solving the following system of equations.

$$\begin{aligned} p_1\psi(s_1, W_1) &= q \\ p_i\psi(s_{i+1}, W_i) &= p_{i+1}\psi(s_{i+1}, W_{i+1}) \quad i = 1, 2., , , K-1 \end{aligned}$$

The first equation above indicates that the least skilled worker who participates only breaks even in expectation. We prove in the appendix that $\psi(s, W_i)$ is monotonically increasing in s and decreasing in i. This means that any worker whose skill is less than s_1 will have negative expected payoff and therefore will not participate. Similarly, by recalling that groups are contiguous, the second equation gives a condition that makes workers at the boundary between successive groups indifferent. Again, from the monotonicity of $\psi(s, W_i)$, we find that an increase in payment p_{i+1} over payment p_i by a factor $\frac{\psi(s_{i+1}, W_i)}{\psi(s_{i+1}, W_{i+1})}$ will provide lower expected payoff to any worker whose skill is less than s_{i+1} who decides to move to group $i+1$, or any worker whose skill is higher than s_{i+1} and decides to move to group i. This is because, upon a move to a group with higher index, the drop in probability of successful completion for a less-skilled worker is not sufficiently offset by the increase in payment, and the opposite is true for a higher-skilled worker who contemplates moving in the opposite direction.

The next example shows how Proposition 3 is used to find prices that implement the hierarchy obtained in Example 1 in a decenteralized fashion.

Example 2. We revisit Example 1 to show how the optimal two group assignment can be implemented. We want to find q, p_1, and p_2 that will provide the correct incentives to workers to participate in the groups they were allocated to in that example. Let us denote by the dummy group W_0 the

group of workers who choose not to participate. Recall that the optimal groups were (in terms of skill) $W_0 = (0, 0.25)$, $W_1 = (0.25, 0.75)$, and $W_2 = (0.75, 1)$. By setting $q = 0.25, p_1 = 1$, and $p_2 = 1.5$, we obtain the desired hierarchy. To see this, let us examine a deviation of a worker of skill s in W_0. This worker makes 0 in that group, and by deviating to join group W_1, he obtains an expected utility of $p_1\psi(s, W_1) - q$. For this example, $\psi(s, W_1) = s$, and so the expected payoff from deviation is equal to $1s - 0.25$, which is less than zero for $s < 0.25$. Similarly, a worker of skill s in W_1 makes an expected profit of $1s - 0.25$, which is higher than 0 but also higher than what he would make had he decided to join W_2 instead. In that case he will be making $p_2\psi(s, W_2) - q = 1.5(s - 0.25) - 0.25$, which is less than $1s - 0.25$ for $s \in (0.25, 0.75)$. Finally, a worker of skill s in W_2 has $\psi(s, W_2) = \frac{1}{2} + \frac{s-0.75}{0.5}$ for an expected payoff of $1.5(\frac{1}{2} + \frac{s-0.75}{0.5}) - 0.25$. This is higher than what a worker whose skill is in $(0.75, 1)$ stands to gain by moving to W_1.

5 Final Remarks

This paper examines the problem of sourcing a range of tasks to a broad set of workers of unknown skills using a decentralized marketplace. In these large and anonymous markets, employers cannot assign workers to specific tasks and do not know the workers' specific skills. In addition, the nature of the tasks to be performed often preclude specific information about how difficult the given task is and what type of workers will be able to complete them.

Our main insight is that despite the informational and coordination constraints that firms face in this setting, these decentralized markets can be organized in a way that makes workers behave as if they are part of an organization, but with some constraints resulting from the market allocation. This crowd organization emerges endogenously as an equilibrium of the incentives offered in the marketplace. The optimal organization involves a sequential matching in which workers are sorted into different groups of increasing skill level, and are successively assigned to tasks with endogenously increasing difficulties (the endogeneity is a consequence of the fact that the tasks assigned to latter groups are those that earlier groups were unable to complete). This application can be supported by a simple pricing mechanism in which prices — rewards —for tasks increase the longer these tasks remain uncompleted. One advantage of this approach is that it automates the burdensome and error-prone process of pricing individual tasks by off-loading it to a market mechanism.

This optimal organization is notable because of its ability to economize on the scarce factor in this marketplace — the time of skilled workers. The effort to economize on these skills is reflected in a range of interesting, and perhaps surprising, results of our framework. First, though the optimal organization is hierarchical, this hierarchy may not look like a pyramid — which contrasts with existing results in organizational economics and on related assignment problems. This happens because sometimes pyramids waste the skills of middle-skill workers as they expose them excessively to difficult tasks which they fail to perform. Second, for similar reasons, increasing the number of layers of the hierarchy may not be beneficial — it may even lead to lower output. This is because more layers will increase the number of workers that are exposed to the most difficult tasks which will waste the skills of many types of workers. However, too few levels of hierarchy is also not optimal, because these will waste the skills of the most skilled workers on easy tasks. We circumvent some of the difficulties associated with the unexpected behavior of these hierarchies by showing that a hierarchy with only two levels, while not necessarily optimal, is enough to generate most of the throughput obtained by the optimal hierarchy.

Our framework can incorporate some natural extensions like discounting and probabilistic completion of tasks, as we discuss below.

Discounting It is possible that tasks that are finished later in the process count for less than those finished earlier. This has no effect on our results. Depending on the discount factor, either the optimal arrangement remains a hierarchy, or the solution is simply a single group that comprises the most skilled workers (i.e. a single-level hierarchy). To see this, consider the case $k = 2$ and recall that the reason a hierarchy emerges as an optimal assignment is that the second group, instead of working on easy tasks, are more focused on working on tasks that are commensurate with their skills. If this solution changes when discounting is introduced in such a way that the least-skilled workers in W_2 are moved to W_1, then the organization is still a hierarchy. But if the first group in an optimal solution contains some workers who are higher-skilled than those in the second group, then this indicates that discounting is high enough that it is not worth the improvement in output we get from placing those high-skilled workers in the second group, and in that case workers who are even less skilled are better utilized by including them in the first group as well (potentially displacing some even less-skilled workers in that group, and in the extreme case, forming a single group of the most skilled

workers), where their output will be higher. This is especially true since some of the tasks that these low-skilled workers are meant to finish will be finished by the higher-skilled group in W_1 anyway, and assigning those less-skilled workers to the second group will only negatively impact their overall contribution. This indicates that a group of workers w will not move from W_2 to W_1 in an optimal solution unless all those workers in W_2 who are less-skilled than w are also moved to W_1, maintaining the hierarchical structure.

Probabilistic Completion of Tasks The model we presented assumes that a worker finishes a task if their skill is at least equal to the difficulty of the task. A more reasonable assumption might be that workers do not always finish tasks that are below their skills with certainty, but rather, that higher-skilled workers have a higher chance of finishing a task compared to their less-skilled peers. This means that for workers i and j with $s_i > s_j$ and a task t, we have $Pr(i \text{ finishes } t) > Pr(j \text{ finishes } t)$ for all t with $d_t \leq s_j < s_i$. This change does not affect Lemmas 1 and 2, and our results carry through in that modified setting as well.

There are other tools that can be used in practice for ensuring an efficient allocation of scarce resources. The first is an explicit tournament, where workers are rewarded for completing tasks before others. This has not been a problem in our formulation because there is no moral hazard or effort decision — in our framework, a worker can either complete a given task or not, and thus it is always better to sequentially assign workers to tasks and not waste the skills of a worker on a task that will already be successfully completed by another worker. An interesting area for theoretical research would be a framework that combines both the hierarchical and tournament approaches. This framework would design these environments similarly to how sports leagues are organized, with several divisions corresponding to different skill and competition levels. One can have multiple tournaments, with the tasks and workers in each tournament selected through a hierarchical mechanism, so that tasks that are not completed in lower tournaments are passed on to higher tournaments and worker who have performed well in lower tournaments advance to higher ones and so on. As noted earlier, TopCoder utilizes a similar structure for its rated events, where participants can move between levels if their performance improves or falls.

Second, in practice, the asymmetry of information between firms and potential workers can be ameliorated if workers are able to build a reputation. Such reputation building may even be possible

in online markets as shown by the creative work of Pallais (2014). However, the difficulty in enforcing liabilities complicates the efforts to build platforms that leverage reputation to improve the assignment of heterogeneous workers to tasks. A dynamic analysis of these issues and work on the design of better platforms are other important areas for research.

Finally, it is important to note that this area would benefit from more detailed empirical evidence on how workers make their decisions of what types of tasks to work on and how firms design incentives, monetary or otherwise. Using experimental methods and different treatments for workers on various online platforms is a possible way of tackling these questions and constitutes yet another area for interesting future research.

Appendix

Preliminaries and Notation. This section starts by presenting a few straightforward lemmas that are used in the proofs of the main results. We will denote different sequential matches by different superscripts, e.g. μ^A and μ^B. Each sequential match consists of individual matches that inherit the superscript of the sequential match it is part of and has a subscript that describes its ordering in the sequential match. We write this as, for example, $\mu^A = (m_1^A, m_2^A, ...)$. Finally, for each individual matching m_j^i, we denote the worker and tasks groups associated with this matching by W_j^i and T_j^i and write this as $m_j^i = (W_j^i, T_j^i)$. This implies that, for example, the set of tasks T_2^i consists of those tasks left over after matching m_1^i, i.e. $T_2^i = \bar{T}_1^i$ (Recall the notation that the set $\bar{T}_1^i$ indicates the tasks not completed in matching m_1^i). The size of a matching $m(W,T)$ is denoted by $\lambda(m) = \min\{\lambda(W), \lambda(T)\}$, since for example, if workers are more than tasks then the matching can be at most of size equal to the size of tasks, and vice versa.

The probability that a task is successfully completed in m is given by $\pi(m)$. Let F_{d_T} and F_{s_W} be the distributions of difficulties and skills in a group of tasks T and a group of workers W, respectively, and consider matching $m(W,T)$, then $\pi(m)$ can be written as

$$\pi(m) = \int_{d_1}^{d_2} (1 - F_{s_W}(\delta)) f_{d_T}(\delta) d\delta,$$

where $[d_1, d_2]$ is the domain of difficulties in T and f_{d_T} is the density of difficulties.

The probability of successful completion is simply the probability that a worker in W is matched with a task in T such that the difficulty of the task is within the worker's skill.

The following elementary fact will be useful at several distinct points in our analysis.

Fact 1. (Task regions) Consider matching $m(W,T)$ and assume for simplicity that $\lambda(W) = \lambda(T)$. Let $s_l = \min\{s_i | i \in W\}$ and $s_h = \max\{s_i | i \in W\}$. The set T can be divided into three regions, R1, R2, and R3, where:

- R1 These are the tasks that can be done by every worker in W. The probability that a task is in R1 is equal to $F_D(s_l)$, and this region has measure $\lambda(T)F_D(s_l)$.
- R2 These are the tasks that can be done by some, but not all, workers in W, and their difficulties lie in (s_l, s_h). The probability that a task is in this region is equal to $F_D(s_h) - F_D(s_l)$, and this region has measure $\lambda(T)(F_D(s_h) - F_D(s_l))$.
- R3 These are the tasks that cannot be completed by any worker in W. The probability that a task is in R3 is equal to $1 - F_D(s_h)$, and this region has measure $\lambda(T)(1 - F_D(s_h))$.

A Proofs

Lemma A.1. Let W be a group of workers, and consider matchings $m = m(W,T)$ and $m' = m(W,T')$, where T and T' are two task pools with $\lambda(T) = \lambda(T')$ and $d_T \succ d_{T'}$, then $\theta(m) \leq \theta(m')$.

Proof: Let $f_W(s)$ be the density function for skills in W, with support over $[s_1, s_2]$, and write the outputs of m and m' as

$$\theta(m) = \lambda(m)\pi(m) = \lambda(m) \int_{s_1}^{s_2} F_{d_T}(s) f_W(s) ds \tag{2}$$

$$\theta(m') = \lambda(m')\pi(m') = \lambda(m) \int_{s_1}^{s_2} F_{d_{T'}}(s) f_W(s) ds \tag{3}$$

Since $d_T \succ d_{T'}$ implies $F_{d_T}(s) \leq F_{d_{T'}}(s)$ for all s and $F_{d_T}(s) < F_{d_{T'}}(s)$ for some s, we have $(2) \leq (3)$ and hence $\theta(m) \leq \theta(m')$. ■

Lemma A.1 states that the output of the same group of workers cannot improve when tasks become more difficult. The next lemma complements this by showing that a group's output on the same tasks can only increase if some workers in the group are replaced by an equal number of higher-skilled workers.

Lemma A.2. Consider matchings $m = m(W, T)$ and $m' = m(W', T)$, where $W' = (W \setminus w) \cup w'$, $w \subset W$, $w' \cap W = \emptyset$, $w' \succ w$, and $\lambda(w) = \lambda(w')$, then $\theta(m) \leq \theta(m')$.

Proof: Denote by S the set of skills in $W \cup w'$. Because $w' \succ w$, there is a set D such that $F_{s_W}(d_T) = F_{s_{W'}}(d_T)$ for $d_T \in D$ and $F_{s_W}(d_T) > F_{s_{W'}}(d_T)$ for $d_T \in \bar{D}$, where $D \cap \bar{D} = \emptyset$ and $D \cup \bar{D} = S$. Letting (d_1, d_2) be the domain of difficulties in T, we can write down the expressions for $\pi(m)$ and $\pi(\bar{m})$ as

$$\pi(m) = \int_{d_1}^{d_2} (1 - F_{s_W}(\delta)) f_{d_T}(\delta) d\delta$$

and

$$\pi(m') = \int_{d_1}^{d_2} (1 - F_{s_{W'}}(\delta)) f_{d_T}(\delta) d\delta$$

It is straightforward to see that, for any $\delta \in (d_1, d_2)$

$$\begin{aligned} F_{s_W}(\delta) &\geq F_{s_{W'}}(\delta) \\ \int_{d_1}^{d_2} (1 - F_{s_W}(\delta)) f_{d_T}(\delta) d\delta &\leq \int_{d_1}^{d_2} (1 - F_{s_{W'}}(\delta)) f_{d_T}(\delta) d\delta \\ \pi(m) &\leq \pi(m') \end{aligned}$$

Since $\theta(m) = \lambda(m)\pi(m)$ and $\theta(m') = \lambda(m')\pi(m')$ and $\lambda(m) = \lambda(m')$, the result follows. ■

Lemma A.3. Consider matchings $m = m(W, T)$, $m' = m(W', T)$, and $m'' = m(W, T')$. Let w and t be a group of workers and a group of tasks such that $W \cap w = \emptyset$, $W' = W \cup w$, $t \subset T$, and $T' = T \setminus t$, then

a. If $\lambda(T) = \lambda(W)$ and w is such that $W \succ w$, then $\theta(m) \geq \theta(m')$.

b. If $\lambda(T) > \lambda(W)$ and w is such that $\lambda(w) = \lambda(T) - \lambda(W)$, then $\theta(m) \leq \theta(m')$.

c. If $\lambda(T) \leq \lambda(W)$, then $\theta(m'') = \theta(m) - \theta(m(W,t))$.

Proof: For part a, note that $W \succ w$ implies $s_W \succeq s_{W'}$, i.e. $F_{s_W} \leq F_{s_{W'}}$, and hence by the same arguments of Lemma A.2 we have $\pi(m) \geq \pi(m') \rightarrow \theta(m) \geq \theta(m')$. For part b, let $\bar{T} \subset T$ be the set of tasks assigned to W in m, then because matchings are uniformly random, the same set of tasks will be assigned to $W \subset W'$ in m', making $\theta(m') \geq \theta(m)$. The same argument applies to part c with the bigger set being W instead of T. ■

Lemma A.3a states that if the number of workers in a group is equal to the number of tasks given to that group, then there is no reason to add any more workers whose skills are less than the skills of everyone else in the group. The reason is that this will cause a redistribution of tasks in a way that results in some tasks going to less-skilled workers instead of the workers they were originally assigned to, while the remaining tasks have their original assignment, and hence the output can only go down. A simple corollary of this claim is that no optimal assignment will assign a group of tasks to a group of workers such that the number of workers is more than the number of tasks.

Corollary A.1. In an optimal solution, no set of tasks T is assigned to a group W such that $\lambda(W) > \lambda(T)$.

In contrast to Lemma A.3a, Lemma A.3b says that extending the size of a group of workers to make it equal to the number of assigned tasks in a matching cannot reduce output. This is because adding more workers in this scenario does not affect the distribution of tasks in the original smaller-sized group. Instead, the effect is that tasks that were not originally assigned due to a mismatch in group size will now have extra workers to attempt them.

Finally, Lemma A.3c states that when the number of workers is at least equal to the number of tasks, removing some tasks from a matching does not affect the output of the group of workers on the remaining tasks.

Proof of Lemma 1: Consider matching $m(W,T)$ and assume for simplicity that $\lambda(W) = \lambda(T)$. We can divide T into three regions as in Fact 1, with s_l and s_h being the lowest and highest skills in W, respectively. Let γ_i be the probability that a worker in W finishes a task in region i, so that $\gamma_1 = 1$ and $\gamma_3 = 0$. From the preliminaries, we know that γ_2 is given by

$$\gamma_2 = \int_{s_l}^{s_h} (1 - F_{s_W}(\delta)) f_{d_T}(\delta) d\delta$$

Recall that $\bar{T}$ denote the tasks left after the matching. The set $\bar{T}$ has two regions:

- Region A: This is the same as R3 in Fact 1, with measure $\lambda(T)(1 - F_{d_T}(s_h))$.
- Region B: This contains the uncompleted tasks from R2, and has measure $(1-\gamma_2)\lambda(T)(F_{d_T}(s_h) - F_{d_T}(s_l))$.

This implies that the relative measure of tasks in R3 has increased. Tasks with difficulties over s_h had a proportion of $1 - F_{d_T}(s_h)$ in T, but in $\bar{T}$, the same tasks now have a proportion

$$\begin{aligned}\frac{\lambda(T)(1-F_{d_T}(s_h))}{\lambda(T)(1-F_{d_T}(s_h))+(1-\gamma_2)\lambda(T)(F_{d_T}(s_h)-F_{d_T}(s_l))} &= \frac{1-F_{d_T}(s_h)}{1-F_{d_T}(s_h)+(1-\gamma_2)(F_{d_T}(s_h)-F_{d_T}(s_l))} \\ &= \frac{1-F_{d_T}(s_h)}{1-F_{d_T}(s_l)-\gamma_2(F_{d_T}(s_h)-F_{d_T}(s_l))} \\ &> 1-F_{d_T}(s_h)\end{aligned}$$

To understand the distribution of the tasks in Region B, note that the probability $1-F_{s_W}(\delta)$ of finishing a task in R2 is decreasing in δ. Consider intervals $d_1 = (\delta_1, \delta_2)$, and $d_2 = (\delta_3, \delta_4)$, where d_1 and d_2 are both in T, with $\delta_3 > \delta_2$ and $\lambda(d_1) = \lambda(d_2)$. The probability of finishing a task in (δ_1, δ_2) is given by

$$\begin{aligned}\gamma &= \int_{\delta_1}^{\delta_2}(1-F_{s_W}(\delta))f_{d_T}(\delta)d\delta \\ &\geq \int_{\delta_3}^{\delta_4}(1-F_{s_W}(\delta))f_{d_T}(\delta)d\delta \\ &= \gamma'\end{aligned}$$

where γ' is the probability of finishing a task in (δ_3, δ_4). Thus the measure of tasks in d_1 decreases by $\gamma\lambda(d_1)$ while the measure of tasks in d_2 decreases by $\gamma'\lambda(d_2)$. For $i = 1, 2$, let $d_i' \subset \bar{T}$ be the tasks in d_i that were not finished in the matching. We have $\frac{\lambda(d_1')}{\lambda(d_2')} = \frac{\lambda(d_1)(1-\gamma)}{\lambda(d_2)(1-\gamma')} = \frac{(1-\gamma)}{(1-\gamma')} \leq 1$ and therefore the relative measure of more difficult tasks in R3 is nondecreasing in $\bar{T}$ compared to T. Let j and j' be two randomly chosen tasks from T and $\bar{T}$, respectively, then for any value of difficulty d, $Pr(d_j \leq d) \geq Pr(d_{j'} \leq d))$ and $Pr(d_j \leq d) > Pr(d_{j'} \leq d))$ for $d > s_h$, implying that $d_{\bar{T}} \succ d_T$. ■

The following lemma derives a relationship between group dominance, group ordering, and output.

Lemma A.4. Let T be a group of tasks and consider two groups of workers W_1 and W_2 with $\lambda(W_1) = \lambda(W_2) \leq \frac{\lambda(T)}{2}$ and such that $W_2 \succ W_1$. Let $\mu^A = (W_1, W_2)$ and $\mu^B = (W_2, W_1)$, then $\theta(\mu^B) \leq \theta(\mu^A)$.

Proof: Let $\lambda(T)$ be normalized to 1 and let $\lambda(W_1) = \lambda(W_2) = l \leq \frac{1}{2}$. Assume wlog that the least difficult task in T and the lowest skill worker in W_1 have value zero. Let $d_1 = \max\{s_i : i \in W_1\}$ and $d_2 = \max\{s_j : j \in W_2\}$, and divide the difficulties in T into intervals $(0, d_1), (d_1, d_2)$, and (d_2, d_{max}), where d_{max} is the maximum difficulty in T. Let

$$F_{d_T}(d) = \begin{cases} x_1 & d = d_1 \\ x_2 & d = d_2 \\ 1 & d = d_{max} \end{cases}$$

Denote by $T_{(\underline{d},\bar{d})}$ the set of tasks in T whose difficulties lie in $(\underline{d}, \bar{d})$ and define matchings $m = m(W_1, T_{(0,d_1)})$

and $m' = m(W_2, T_{(d_1,d_2)})$. Recall that the probabilities of finishing a task in m and m' are given by $\pi(m)$ and $\pi(m')$, respectively. Because $W_2 \succ W_1$, $Pr(s_j \geq d) = 1$ for all $j \in W_2$ and $d \in (0, d_1)$. Now define $\mu^A = (m_1^A, m_2^A)$, with $m_1^A = m(W_1, T)$ and $m_2^A = m(W_2, T_2^A)$, where T_2^A are the tasks left over after the matching m_1^A. Similarly, let $\mu^B = (m_1^B, m_2^B)$, with $m_1^B = m(W_2, T)$ and $m_2^B = m(W_1, T_2^B)$, where T_2^B are the tasks left over after the matching m_1^B. We have $\theta(m_1^A) = \theta(m) = lx_1\pi(m)$ and $\lambda(T_2^A) = 1 - lx_1\pi(m)$, with $d_{T_2^A}$ distributed as

$$F_{d_{T_2^A}}(d) = \begin{cases} \frac{x_1(1-l\pi(m))}{1-lx_1\pi(m)} & d = d_1 \\ \frac{x_2-lx_1\pi(m)}{1-lx_1\pi(m)} & d = d_2 \\ 1 & d = d_{max} \end{cases}$$

and hence $\theta(m_2^A) = l\left(\frac{x_1(1-l\pi(m))+\pi(m')(x_2-x_1)}{1-lx_1\pi(m)}\right)$. This means that

$$\theta(\mu^A) = \theta(m_1^A) + \theta(m_2^A) = lx_1\pi(m) + l\left(\frac{x_1(1-l\pi(m)) + \pi(m')(x_2-x_1)}{1-lx_1\pi(m)}\right) \tag{4}$$

Proceeding in the same fashion, we compute $\theta(m_1^B) = l(x_1 + (x_2 - x_1)\pi(m'))$. The probability that a task has difficulty in $(0, d_1)$ in T_2^B is given by

$$F_{d_{T_2^B}}(d_1) = \frac{x_1(1-l)}{1 - l(x + \pi(m')(x_2 - x_1))}$$

Consequently, $\theta(m_2^B) = l\pi(m)\left(\frac{x_1(1-l)}{1-l(x+\pi(m')(x_2-x_1))}\right)$, and

$$\theta(\mu^B) = \theta(m_1^B) + \theta(m_2^B) = l(x_1 + (x_2 - x_1)\pi(m')) + l\pi(m)\left(\frac{x_1(1-l)}{1 - l(x + \pi(m')(x_2 - x_1))}\right) \tag{5}$$

Since $d_2 > d_1$ implies $x_2 > x_1$, it can be verified that (4) is indeed greater than or equal to (5) for any values of l, $\pi(m)$, and $\pi(m')$, and hence $\theta(\mu^A) \geq \theta(\mu^B)$.

■

Lemma A.4 indicates that it is optimal to arrange groups with dominance relationships in ascending order, with dominant groups placed later in the process. Lemma A.5 formalizes the idea that an increase in the skill pool of the first group can only lead to the workers in the second group being worse off in terms of output, but that the overall output is improved if the workers with improved skills are still less-skilled than those in the second group.

Lemma A.5. Let T be a group of tasks and W_1 and W_2 be groups of workers with $W_2 \succ W_1$. Consider $w \in W_1$ and $w' \notin W_1$, with $\lambda(w) = \lambda(w')$, $w' \succ w$, and $W_2 \succ w'$. Define $W_1' = (W_1 \setminus w) \cup w'$ and matchings $\mu^A = (m_1^A, m_2^A)$ with $m_1^A = m(W_1, T)$ and $m_2^A = m(W_2, T_2^A)$, and $\mu^B = (m_1^B, m_2^B)$ with $m_1^B = m(W_1', T)$ and $m_2^B = m(W_2, T_2^B)$, then $\theta(m_2^A) \geq \theta(m_2^B)$ and $\theta(\mu^B) \geq \theta(\mu^A)$.

Proof: Consider a worker with skill s in W_2 and define $T_2^i(s) = \{t \in T_2^i | d_t \leq s\}$ for $i = A, B$, i.e. $T_2^i(s)$

is the set of tasks in T_2^i that a worker with skill s can finish. Note that w and w' will be matched to the same group of tasks in m_1^A and m_1^B, respectively. Because $w' \succ w$, the amount of finished tasks in m_1^A, denoted by $\lambda(T^c_{m_1^A})$, is less than $\lambda(T^c_{m_1^B})$, the amount of finished tasks in m_1^B. This implies that $T_2^B(s) \subset T_2^A(s)$, and hence $\lambda(T_2^B(s)) = \lambda(T_2^A(s)) - \tau$ and $\lambda(T^c_{m_1^A}) = \lambda(T^c_{m_1^B}) - \tau$, for some $\tau \geq 0$. The probability that a worker with skill s gets a task that he can finish in m_2^B is equal to

$$\begin{aligned}\frac{\lambda(T_2^B(s))}{\lambda(T) - \lambda(T^c_{m_1^B})} &= \frac{\lambda(T_2^A(s)) - \tau}{\lambda(T) - \lambda(T^c_{m_1^A}) - \tau} \\ &\leq \frac{\lambda(T_2^A(s))}{\lambda(T) - \lambda(T^c_{m_1^A})},\end{aligned}$$

where the right hand side of the equality is the probability that a worker with skill s can finish a task in m_2^A, i.e. the proportion of tasks that W_2 can finish has shrunk, and W_2 now finishes a smaller proportion from a smaller set of tasks in m_2^B compared to m_2^A, and therefore $\theta(m_2^A) \geq \theta(m_2^B)$.

To show that $\theta(\mu^B) \geq \theta(\mu^A)$, consider the same worker with skill s in W_2 who is assigned a task $j \in T_2^A$ in m_2^A. If $j \in T_2^A$ and $j \notin T_2^B$, then this implies that j is already finished in m_1^B and in addition this worker is assigned a positive share of tasks in $T_2^B(s)$ (with a ratio that is equal to their proportion in T_2^B), leading to a combined overall increase in output.

■

We use Lemmas A.4 and A.5 to prove the following lemma in preparation for proving Lemma 2. Roughly, Lemma A.6 states that any improvement on a two-group hierarchical arrangement can only be hierarchical, by either moving some of the most skilled workers from the first group to the second, or moving some of the least skilled workers from the second group to the first.

Lemma A.6. Let W_1 and W_2 be two worker groups such that $W_2 \succ W_1$ and denote by μ the matching $(m_1 = m(W_1, T), m_2 = m(W_2, T_2))$.

a. Consider w and $w' \subset W_1$ such that $w \succ w'$ and $\lambda(w) = \lambda(w')$, and let $\mu^A = (m_1^A, m_2^A)$ with $m_1^A = m(W_1 \setminus w, T), m_2^A = m(W_2 \cup w, T_2^A)$, and $\mu^B = (m_1^B, m_2^B)$ with $m_1^B = m(W_1 \setminus w', T), m_2^B = m(W_2 \cup w', T_2^B)$. If $\theta(\mu^B) \geq \theta(\mu)$, then $\theta(\mu^A) \geq \theta(\mu^B)$.

b. Consider w and $w' \subset W_2$ such that $w \succ w'$ and $\lambda(w) = \lambda(w')$, and let $\mu^A = (m_1^A, m_2^A)$ with $m_1^A = m(W_1 \cup w, T), m_2^A = m(W_2 \setminus w, T_2^A)$ and $\mu^B = (m_1^B, m_2^B)$ with $m_1^B = (W_1 \cup w', T), m_2^B = m(W_2 \setminus w', T_2^B)$. If $\theta(\mu^A) \geq \theta(\mu)$, then $\theta(\mu^B) \geq \theta(\mu^A)$.

Proof: To prove Part a, assume as in the statement of the lemma that $\theta(\mu^B) \geq \theta(\mu)$. We will denote by $W_1^A = W_1 \setminus w$ and by $W_1^B = W_1 \setminus w'$. First, note that the group of workers $W_1 \setminus (w \cup w')$ has the same output in both m_1^A and m_1^B. Second, we know from Lemma A.4 that w' followed by w has higher output than w followed by w'. Finally, because $w \succ w'$, by Lemma A.5, we get that the output of $W_2 = W_2^A \setminus w$ in m_2^A is greater than the output of $W_2 = W_2^B \setminus w'$ in m_2^B. Combining all three observations, we get that $\theta(\mu^A) \geq \theta(\mu^B)$. The argument for Part b is similar. ■

It is important to note that Lemma A.6 applies when removing several, possibly disjoint intervals of workers simultaneously from one group to another. For example, consider two intervals w_1 and w_2 of workers in W_1 and assume that there is an improvement in output when these intervals are moved to group W_2, then it is always at least as good to instead remove a measure w of workers from W_1 to W_2 with w being the most skilled workers in W_1 whose measure is equal to $\lambda(w) = \lambda(w_1) + \lambda(w_2)$. The same idea holds when removing intervals from W_2 to W_1. This observation along with Lemma A.6 are used in the following proof.

Proof of Lemma 2: As in the statement of the lemma, assume that W_1 and W_2 are not hierarchical. Construct W_1' and W_2' such that $i \in (W_1 \cup W_2) \to i \in (W_1' \cup W_2')$, $\lambda(W_1) = \lambda(W_1')$, $\lambda(W_2) = \lambda(W_2')$, and $W_2' \succ W_1'$. There exists $w_1' \subset W_1'$ and $w_2' \subset W_2'$ –not necessarily continuous intervals– such that $W_1 = (W_1' \setminus w_1') \cup w_2'$ and $W_2 = (W_2' \setminus w_2') \cup w_1'$. Thus, by moving w_2' to W_1' and moving w_1' to W_2, we can reconstruct W_1 and W_2. However, by Lemma A.6, unless either: a) $w_1' = \emptyset$ and w_2' is such that $(W_2' \setminus w_2') \succ w_2'$ or b) $w_2' = \emptyset$ and w_1' is such that $w_1 \succ (W_1' \setminus w_1')$ – either of which imply that W_1 and W_2 are hierarchical and hence contradict our assumption– there exists $\bar{W}_1$ and $\bar{W}_2$ such that $\bar{W}_1$ and $\bar{W}_2$ are hierarchical and $\theta(\bar{W}_1, \bar{W}_2) = \theta(\bar{\mu}) \geq \theta(W_1, W_2) = \theta(\mu)$, where $\theta(A, B)$ denotes the output of a sequential matching that first assigns tasks to group A, then to group B. ■

The optimal groups are contiguous:

Lemma A.7. Consider the optimal grouping $W^* = (W_1^*, W_2^*)$ for $k = 2$, and a worker i with skill s_i. If $i \in W^*$, then $\{j | s_j > s_i\} \in W^*$.

Proof: Assume that there is $w_3 \succ w_2 \succ w_1$ such that $w_1, w_3 \in W^*$, $w_2 \notin W^*$, and $\lambda(w_1) = \lambda(w_2)$. Note that by Lemma A.2, swapping w_1 with w_2 can only improve the output of the group that w_1 is in. This implies that if $w_1 \in W_2^*$ then swapping it out for w_2 can only increase the combined output, which contradicts the optimality of W^*. We then need to note that by Lemma A.5, if $w_1 \in W_1^*$ and $w_3 \in W_2^*$, or if both w_1 and w_3 are in W_1^*, then swapping w_1 out for w_2 can still only improve the combined output and again W^* cannot be optimal. Taking the limit as the sizes of w_1 and w_2 go to zero proves the statement of the lemma. ■

Proof of Proposition 1: We prove the proposition by induction on the number of groups. Define $G(s) : \mathbb{R} \to [0, \lambda(\mathcal{W})]$ to be the measure of workers whose skill is less than or equal to s and let $G^{-1}(w)$ be its inverse, i.e., $G^{-1}(w)$ is the skill at or below which a measure w of workers reside.[3] Lemma 2 shows that the base case for $k = 2$ holds. Assume that the result holds for k and consider $k + 1$ groups. By the induction hypothesis, the optimal solution for the assignment of the remaining tasks from the matching $m(W_1, \mathcal{T})$ is a hierarchy. Furthermore, Lemma A.7 applies to k groups since the

[3]The difference between $F_{s_W}(s)$ and $G(s)$ is that the first is a probability distribution, while the second is not since $\lambda(\mathcal{W}) > 1$.

output of a group of workers can only increase by substituting a dominated subgroup within a group by a group of equal size that dominates it but is not in the current selection of workers. This implies that the optimal solution for k groups starts with a worker with skill $\bar{s}$ and includes all workers up to the maximum skill in $\mathcal{W}$. Therefore, the optimal solution for $k+1$ groups uses at most one more measure of workers than the solution for k groups. This measure consists of workers whose skill lie in $(G^{-1}(G(\bar{s})-1), \bar{s})$. Assume that the optimal solution uses workers whose skills is less than $G^{-1}(G(\bar{s})-1)$, then by Lemma A.2 we can replace these workers by others whose skills are in $(G^{-1}(G(s)-1), \bar{s})$ to improve output. Let W_1 be the first group in the optimal $k+1$ groups solution. Groups W_2 through W_{k+1} form a hierarchy by the induction hypothesis. Consider moving $w \in W_i, i > 2$ to group W_1. By Lemmas A.4 and A.5, this will decrease output. Conversely, by Lemmas 1 and A.1, removing any workers from W_1 to later groups would decrease their output and exchanging them with any of the higher-skilled workers in one of those groups is again output decreasing. By similar arguments to Lemma A.6, the only move with a potential for output maximization is one that removes a group w' from W_2 such that $(W_2 \setminus w') \succ w'$, and adds it to W_1, with possible adjustments to later groups (again, removing workers from the peripheries only), and hence the optimal $k+1$ groups solution is also hierarchical. ■

Lemma A.8. (Monotonicity of success probabilities) For all groups i and all skills s, $\psi(s, W_i)$ is monotonically decreasing in i and monotonically increasing in s.

Proof: Let the distribution of difficulties in group i be given by $F_{d_{T_i}}$ and consider group $j > i$ with difficulty distribution $F_{d_{T_j}}$. A worker with skill s has $\psi(s, W_i) = F_{d_{T_i}}(s)$ in group i and $\psi(s, W_j) = F_{d_{T_j}}(s)$ in group j. By Lemma 1, $d_{T_j} \succeq d_{T_i}$ and therefore $F_{d_{T_i}}(s) \geq F_{d_{T_j}}(s)$, and $\psi(s, W_i)$ is monotonically decreasing in group index, and since F is a distribution function, it is monotonically increasing in its argument s.

■

Corollary A.2. (Monotonicity of output) Denote by $\theta(s, i)$ the output of a worker with skill s in group W_i. In the optimal hierarchical solution $\theta(s, i) \geq \theta(s, j)$ for $j > i$.

Proof of Lemma 3 Consider the optimal grouping and examine the i^{th} matching $m(W_i, T_i)$. Let $\lambda(m)$ be normalized to 1 and assume wlog that the least difficult task in T and the lowest skill in W_i are normalized to zero. Let s_1 and s_2 be the skills of any two workers in W, with $s_2 > s_1$. Divide the difficulties in T into intervals $(0, s_1), (s_1, s_2)$, and (s_2, s_{max}), where s_{max} is the maximum difficulty in T. Let $F_{d_T}(s_1) = x_1$, $F_{d_T}(s_2) - F_{d_T}(s_1) = x_2$, $F_{s_{W_i}}(s_1) = l_1$, and $F_{s_{W_i}}(s_2) - F_{s_{W_i}}(s_1) = l_2$. Denote by π_1 the probability of finishing a task that is randomly assigned from $(0, s_1)$ to a worker whose skill is in $(0, s_1)$. Similarly, denote by π_2 the corresponding probability when a task with difficulty in (s_1, s_2) is assigned to a worker whose skill is in (s_1, s_2). Let T_{i+1} be the set of tasks remaining after $m(W_i)$

and let the measure of tasks in (s_2, s_{max}) in T_{i+1} be equal to y (these are the tasks that no worker in W_i can finish). We have $\lambda(T_{i+1}) = y + x_1(l_1(1-\pi_1)) + x_2(l_1 + l_2(1-\pi_2))$. The distribution of T_{i+1} is therefore given by

$$F_{d_{T_{i+1}}}(d) = \begin{cases} \frac{x_1 l_1(1-\pi_1)}{y+x_1(l_1(1-\pi_1))+x_2(l_1+l_2(1-\pi_2))} & d = s_1 \\ \frac{x_1(l_1(1-\pi_1))+x_2(l_1+l_2(1-\pi_2))}{y+x_1(l_1(1-\pi_1))+x_2(l_1+l_2(1-\pi_2))} & d = s_2 \\ 1 & d = s_{max} \end{cases}$$

Using $F_{d_{T_{i+1}}}$ we can write

$$\phi(s_1) = \frac{\psi(s_1, W_i)}{\psi(s_1, W_{i+1})} = x_1 \frac{y + x_1(l_1(1-\pi_1)) + x_2(l_1 + l_2(1-\pi_2))}{x_1 l_1(1-\pi_1)}$$

and

$$\phi(s_2) = \frac{\psi(s_2, W_i)}{\psi(s_2, W_{i+1})} = (x_1 + x_2) \frac{y + x_1(l_1(1-\pi_1)) + x_2(l_1 + l_2(1-\pi_2))}{x_1(l_1(1-\pi_1)) + x_2(l_1 + l_2(1-\pi_2))}$$

Finally, we have

$$\frac{\phi(s_2)}{\phi s_1} = \frac{(x_1 + x_2) l_1(1-\pi_1)}{x_1(l_1(1-\pi_1)) + x_2(l_1 + l_2(1-\pi_2))} < 1,$$

i.e. $\phi(s_2) < \phi(s_1)$, as required, and the result obviously holds if $i+1$ is replaced by any $j > i+1$. ■

Proof of Proposition 3 Denote by W_0 the group that includes workers whose skills are below the minimum skill for participation $\bar{s}$. Let $s_i = \{\min s | s \in W_i^*\}$, so that $s_1 = \bar{s}$. Consider an entry fee q and payments $p_1, ...p_k$ that satisfy the following system of equations

$$\begin{aligned} p_1 \psi(s_1, W_1^*) &= q \\ p_i \psi(s_{i+1}, W_i^*) &= p_{i+1} \psi(s_{i+1}, W_{i+1}^*) \quad i = 1, 2., , , k-1 \end{aligned}$$

The minimum payoff that can be obtained in W_1^* is equal to $p_1 \psi(s_1, W_1^*) - q = 0$. By Lemma A.8, $\psi(s', W_1^*) < \psi(s_1, W_1^*)$ for all $s' < s_1 = \bar{s}$, and therefore $p_1 \psi(s', W_1^*) - q < 0$. Similarly, for $s > \bar{s}$, $\psi(s, W_1^*) > \psi(\bar{s}, W_1^*)$ and $p_1 \psi(s, W_1^*) - q > p_1 \psi(s_1, W_1^*) - q \geq 0$. Consider a worker with skill $s < s_{i+1}$ in W_i^*. This worker makes $p_i \psi(s, W_i^*)$ and upon moving to W_{i+1}^* makes

$$\begin{aligned} p_{i+1} \psi(s, W_{i+1}^*) &= p_i \frac{\psi(s_{i+1}, W_i^*)}{\psi(s_{i+1}, W_{i+1}^*)} \psi(s, W_{i+1}^*) \\ &= p_i \psi(s, W_i^*) \frac{\psi(s_{i+1}, W_i^*)}{\psi(s_{i+1}, W_{i+1}^*)} \frac{\psi(s, W_{i+1}^*)}{\psi(s, W_i^*)} \\ &= p_i \psi(s, W_i^*) \frac{\phi(s_{i+1})}{\phi(s)} \\ &\leq p_i \psi(s, W_i^*) \end{aligned}$$

where the last inequality is obtained from Lemma 3 and the fact that $s < s_{i+1}$. The inequality still holds if we replace $i+1$ with any $j > i$. A symmetrical argument shows that $p_{i+1}\psi(s, W^*_{i+1}) \geq p_i\psi(s, W^*_i)$ for any $s > s_{i+1}$. ■

References

Allon, Gad and Awi Federgruen (2005), "Outsourcing service processes to a common service provider under price and time competition."

Araman, Victor F and Rene Caldentey (2016), "Crowdvoting the timing of new product introduction." *Available at SSRN 2723515*.

Armony, Mor and Amy R Ward (2013), "Blind fair routing in large-scale service systems with heterogeneous customers and servers." *Operations Research*, 61, 228–243.

Ashlagi, Itai and Peng Shi (2015), "Optimal allocation without money: An engineering approach." *Management Science*, 62, 1078–1097.

Babich, Volodymyr, Simone Marinesi, and Gerry Tsoukalas (2018), "Does crowdfunding benefit entrepreneurs and venture capital investors?"

Beggs, Alan W (2001), "Queues and hierarchies." *The Review of Economic Studies*, 68, 297–322.

Boudreau, K.J., N. Lacetera, and K.R. Lakhani (2011), "Incentives and problem uncertainty in innovation contests: An empirical analysis." *Management Science*, 57, 843–863.

Chen, N., N. Immorlica, A. Karlin, M. Mahdian, and A. Rudra (2009), "Approximating matches made in heaven." *Automata, Languages and Programming*, 266–278.

Dulleck, U., R. Kerschbamer, and M. Sutter (2011), "The economics of credence goods: An experiment on the role of liability, verifiability, reputation, and competition." *American Economic Review*, 101, 526–555.

Faridani, S., B. Hartmann, and P.G. Ipeirotis (2011), "Whats the right price? pricing tasks for finishing on time." In *Proc. of AAAI Workshop on Human Computation*.

Feldman, J., A. Mehta, V. Mirrokni, and S. Muthukrishnan (2009), "Online stochastic matching: Beating 1-1/e." In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, 117–126, IEEE.

Fuchs, William, Luis Garicano, and Luis Rayo (2015), "Optimal contracting and the organization of knowledge." *The Review of Economic Studies*, 82, 632–658.

Gans, Noah, Ger Koole, and Avishai Mandelbaum (2003), "Telephone call centers: Tutorial, review, and research prospects." *Manufacturing & Service Operations Management*, 5, 79–141.

Garicano, L. (2000), "Hierarchies and the organization of knowledge in production." *Journal of Political Economy*, 108, 874–904.

Huang, Yan, Param Vir Singh, and Kannan Srinivasan (2014), "Crowdsourcing new product ideas under consumer learning." *Management Science*, 60, 2138–2159.

Kamenica, E. and M. Gentzkow (2011), "Bayesian persuasion." *American Economic Review*, 101.

Koopmans, Tjalling C and Martin Beckmann (1957), "Assignment problems and the location of economic activities." *Econometrica: journal of the Econometric Society*, 53–76.

Lobel, Ilan, Evan D Sadler, and Lav R Varshney (2015), "Customer referral incentives and social media." *Forthcoming, Management Science*.

Manshadi, Vahideh H, Shayan Oveis Gharan, and Amin Saberi (2012), "Online stochastic matching: Online actions based on offline statistics." *Mathematics of Operations Research*, 37, 559–573.

Moldovanu, B. and A. Sela (2001), "The optimal allocation of prizes in contests." *American Economic Review*, 542–558.

Pallais, Amanda (2014), "Inefficient hiring in entry-level labor markets." *The American Economic Review*, 104, 3565–3599.

Prescott, Edward C and Michael Visscher (1980), "Organization capital." *The Journal of Political Economy*, 446–461.

Retelny, Daniela, Sébastien Robaszkiewicz, Alexandra To, Walter S Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S Bernstein (2014), "Expert crowdsourcing with flash teams." In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 75–85, ACM.

Roy, Andrew Donald (1951), "Some thoughts on the distribution of earnings." *Oxford economic papers*, 3, 135–146.

Sattinger, Michael (1975), "Comparative advantage and the distributions of earnings and abilities." *Econometrica: Journal of the Econometric Society*, 455–468.

Terwiesch, Christian and Yi Xu (2008), "Innovation contests, open innovation, and multiagent problem solving." *Management science*, 54, 1529–1543.

Teulings, Coen N (1995), "The wage distribution in a model of the assignment of skills to jobs." *Journal of Political Economy*, 280–315.

Tinbergen, Jan (1974), "Substitution of graduate by other labor." *Kyklos*, 27, 217–226.

Valentine, Melissa A, Daniela Retelny, Alexandra To, Negar Rahmati, Tulsee Doshi, and Michael S Bernstein (2017), "Flash organizations: Crowdsourcing complex work by structuring crowds as organizations." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3523–3537, ACM.

Wallace, Rodney B and Ward Whitt (2005), "A staffing algorithm for call centers with skill-based routing." *Manufacturing & Service Operations Management*, 7, 276–294.

Wolinsky, A. (1993), "Competition in a market for informed experts' services." *The RAND Journal of Economics*, 380–398.