Repeated Matching Mechanism Design with 
Moral Hazard and Adverse Selection

Kartik Ahuja, Mihaela van der Schaar,

University of California, Los Angeles

February 9, 2016

Abstract

In crowdsourcing systems, the mechanism designer aims to repeatedly match a set of tasks of varying qualities (high quality tasks generate high revenue), which are known to the designer, to a set of agents of varying qualities (high quality agents generate high revenue), which are unknown to the designer, such that the overall system performance (e.g. revenue) is maximized. However, in any realistic system, the designer can only observe the output produced by an agent, which is stochastic, and not the actual quality of the agent. (Agents themselves may not know their own qualities.) Thus, the designer needs to learn the quality of the agents and solve an adverse selection problem. Moreover, the expected values of agents’ outputs depend not only on the qualities of the tasks and the qualities of the agents, but also on the efforts exerted by the agents. This is because agents are strategic and want to optimally balance the rewards and costs of exerting effort. Hence, the designer needs to simultaneously learn about the agents and solve a joint moral hazard and adverse selection problem. In this paper we develop a first mechanism that learns and repeatedly matches agents to tasks in a manner that mitigates both adverse selection and moral hazard. We compute the equilibrium strategies for the agents that have a simple bang-bang structure and also enable the agents to learn their qualities. We prove that this mechanism achieves in equilibrium high long-term performance in terms of profit and output.
1 Introduction

In recent years a variety of crowdsourcing systems have emerged. These systems have been used in a variety of settings: in machine learning for labelling tasks [Shah et al. (2015)], in large scale educational settings such as Massive Open Online Courses (MOOCs) for grading homeworks [Diez et al. (2013)], in on-demand economy such as TopCoder for designing software. Generally, the designer needs to repeatedly match the set of tasks to the available pool of agents. The tasks that are available to the designer differ in quality. High quality tasks generate more revenue. The agents participating in the crowdsourcing system also differ in quality. High quality agents generate more revenue. In addition to this, the agents are strategic and differ in the effort they exert, where we identify effort with the time spent on the task.

The first part of the paper constructs a mechanism that repeatedly matches agents with tasks and rewards them based on their performance (i.e. the revenue generated). The mechanism maximizes the long-term performance (for e.g, the long-term profit) while considering the following dimensions:

1. **Adverse selection:** The qualities of the agents are not known to the designer and cannot be observed. The designer only observes the outputs produced. The mechanism needs to ensure that the designer learns the qualities through the output. This is non-trivial since the agent’s output depends on the quality of the agent itself and the effort exerted.
2. **Moral hazard:** The agents are foresighted and only exert high effort if it leads to long-term benefits for themselves. The mechanism needs to optimally balance the incentives provided against the profits of the designer.
3. **Local computability:** The agents have incomplete information about the environment (i.e other agents utilities, qualities and efforts). Hence, the mechanism needs to ensure that the agents can compute their equilibrium strategy locally, using solely the information they possess.
4. **Computational complexity:** Most crowdsourcing systems involve a large number of agents and a large number of tasks. Hence, the mechanism needs to have a low complexity that scales reasonably with the number of agents and the tasks.
5. **Privacy:** To preserve the privacy of the agents, the mechanism needs to ensure that the output history and payments are kept private and not revealed
to others.

In our mechanism the designer decides and commits to a matching function and a payment function, then announces it to the agents. The proposed matching function requires that in the first time slot (which we refer to as the ranking stage) the designer matches the agents to the tasks using a fixed matching rule (chosen randomly since the designer has no knowledge yet about agent’s qualities or effort they exert). The designer then estimates the output per unit task quality (which we refer to as normalized output) and ranks the agents based on their normalized outputs. In all the future time slots the designer matches the agents and tasks with the same rank, where the ranking of the tasks is based on their quality. The payment per unit output made to the agents increases linearly with the normalized output. For the above matching and payment function we compute a Nash equilibrium (NE) strategy for the agents. The utility achieved by the agents by following the proposed strategy is the highest it can achieve in any NE. We also show that for the proposed strategy both the total long-term output and the long-term profits for the designer are high. In addition to this, the proposed strategy has two important features: it has a simple bang-bang structure (exert maximum effort possible or zero effort) and is locally computable.

In the second part of this paper we consider a stochastic setting where the outputs produced by the agents depend stochastically on the quality of the agent, quality of the task, and the effort exerted. We also relax the assumption that agents know their own qualities. The proposed matching mechanism will start with a ranking stage. In the ranking stage, at the start of each time slot, the designer matches the agents based on some fixed matching rule (chosen randomly). It develops an estimate for the mean normalized outputs of the agents by observing the stochastic normalized outputs in multiple time slots. In all the future time slots the designer matches the agents and tasks with the same ranks. The payment function per unit output made to the agents increases linearly with normalized output. Given the matching and payment function, the agents strategically learn their own qualities and decide how much effort to exert in each time slot taking their long-term utility into consideration. We show that if all the agents follow the proposed learning strategy, then it is a NE and each agent achieves the highest possible utility that can be achieved in any NE. Similar to the first part, we prove that the proposed learning strategy has a bang-bang structure and is locally computable. We also show that the proposed matching and payment function can lead to high long-term profit and high total long-term outputs.
2 Related Literature

**Literature on Crowdsourcing:** In Tran-Thanh et al. (2012) the task allocation problem is formulated using bounded multi-armed bandits. The designer has a limited budget and must allocate tasks and learn agent qualities, however the agents are not strategic in this setting (i.e. no moral hazard). In Dayama et al. (2015) the model considers strategic agents that may lie in reporting their costs. This requires a mechanism design approach that can enforce truthful reporting. The key difference between this setting and ours is that: we do not know the quality of the agents, and the cost to completion is dependent on the agents exerted effort, not just on task completion. In Ho et al. (2012) the agents are strategic about exerting effort and hence, there is moral hazard. However, the agents have a public reputation (i.e., no privacy) which is updated based on a social norm. Additionally, the matching of the agents is random hence no adverse selection. In Ho et al. (2014) the setting that is considered assumes that the designer deals with agents of unknown qualities (adverse selection) and exert unobservable effort (moral hazard). A method is presented to learn how to adapt the contract to pay the agents as the designer learns about the distribution of the quality of the agents. In their setting there is no differentiation in the task qualities and hence, the problem of matching or task allocation does not arise. Additionally, the agents in their setting work for one time slot and thus are myopic. Alternatively, the agents in our setting are foresighted and are matched repeatedly.

**Literature on Matching:** There exists a large literature on matching, which can be categorized into two classes: matching without transfers that starts with the seminal work Gale & Shapley (1962) and matching with transfers that starts with the seminal work Shapley & Shubik (1971). Since our work falls in the latter class we position our work only in relation to this class only. Matching with transfers has been applied to numerous areas: to marriage markets Becker (1974), to labor markets Shimer & Smith (2000), to perfect competition Greetsky et al. (1999), to international trade Grossman et al. (2013). These works focus on “proper” matching and assume that the qualities are observable. Hence, adverse selection does not arise and is not addressed in these works. These works take the output as fixed and do not model it as a function of the effort exerted (i.e. they ignore moral hazard), which is a key issue in crowdsourcing applications.
3 Repeated matching mechanism design in a non-stochastic setting

In this section we describe the model for the first part of the paper (the non-stochastic setting). We will enrich this model to incorporate stochasticity in the next section.

**Quality distribution of agents:** Consider a set of $N$ agents $\mathcal{P} = \{1, ..., N\}$ and a designer. Each agent works to produce output. The output of the agent depends among other factors, which are defined later, on its intrinsic quality. $F : \mathcal{P} \rightarrow [0, \infty)$ maps the index of each agent to its intrinsic quality level. We assume that $F$ is injective. Each agent knows its own quality. We will relax this assumption in the second part of the paper. The designer does not know the qualities of the agents, which means the designer faces the problem of adverse selection.

**Quality distribution of tasks:** Consider a set of $N$ tasks $\mathcal{S} = \{1, ..., N\}$ Each agent is typically assigned a task and the output not only depends on the agent’s intrinsic quality but also on the task’s quality. $G : \mathcal{S} \rightarrow [1, \infty)$ maps the index of each task to the quality level. We assume that $G$ is a strictly increasing function. The designer is responsible for assigning the tasks. It also has perfect knowledge of $G$.

**Effort and output of agents:** Each agent needs to exert effort on the task that is assigned to it. If the agent $i$ who is assigned to task $j$ chooses to exert an effort $e_i$, then the total output is proportional to the quality of the agent $F(i)$, the quality of the task $G(j)$ and the effort exerted by the agent $e_i$ and is given as $F(i)G(j)e_i$. We assume that the total output is observed by the designer who in turn informs the agent about it. We can justify this as follows. Assume that the revenue generated per unit output is $b$. The revenue generated from the output of agent $b \left[F(i)G(j)e_i\right]$ is observed by the designer. We define the normalized output $R_i$ as the output produced by the agent per unit task quality, where $R_i = F(i)e_i$. It costs agent $i c_i e_i^2$ to exert effort $e_i$, where $c_i$ is the cost factor that depends on

---

1 The model and the results can be shown to extend when the number of tasks and agents is not the same.

2 We assume without loss of generality that worst task has a non-zero quality of 1 as no agent will work on zero quality task.
the agent. We assume that agent $i$ has an upper limit on the maximum effort that it can exert given by $e_i^{\text{max}} < \infty$. The effort exerted by an agent is only known privately to the agent. We assume that the output is linear in effort and the cost is quadratic to avoid complicated notations but the model and the results can be adapted to more general output and cost functions.

**Time:** We will consider a discrete time model with each time step $t \in \mathcal{T} = \{0, 1, \ldots, \infty\}$. The designer needs to assign to each agent a task in every period. We assume that the quality of the tasks and the quality of the agents do not change with time.

In our mechanism we assume that the designer commits to a matching and a payment function, which we define next, and then announce it to the agents. The public knowledge of payment function is perfectly reasonable (typically on crowdsourcing platforms the agents know how they will be paid upon doing a task). In general it may be possible that the designer does not commit to a matching function. This is the reason why we will compare the performance of our proposed mechanism to benchmarks that will serve as an upper bound to any approach the designer may use including approaches where the designer does not commit.

**Matching function:** In each time step the designer needs to match the agents with the tasks. The designer matches the agents based on the set of observations. We write the set of observations of the designer up to time $t - 1$ (end of time slot $t - 1$) as $h_t^0$, where $h_t^0$ is basically the set of outputs of the agents and the set of the normalized outputs. We assign $h_0^0 = \phi$. We denote the set of all the possible histories up to time $t$ as $\mathcal{H}_0(t)$ and the set of all the possible histories as $\mathcal{H}_0 = \bigcup_{t=0}^{\infty} \mathcal{H}_0(t)$. The matching function is given as $M : \mathcal{H}_0 \times \mathcal{P} \rightarrow \mathcal{S}$; it maps for each agent $i$ and for each history of outputs $h_t^i$ to a task. We will only consider the matching rules that assign workers to tasks entirely based on the output history (independent of agent’s names)$^3$.

**Payment function:** In each time step the designer needs to pay the agents based on the output. The payment function is given as $P : \mathcal{H}_0 \times \mathcal{P} \rightarrow \mathbb{R}$; it maps every agent for every history of output to a payment value per unit output. At the beginning of time slot $t + 1$, agent $i$ receives a payment per unit output denoted

$^3$If two histories of the designer are permutations of each other, then the two matchings will also be the same permutation of each other
as \( P(h_{0}^{t+1}, i) \) for the output produced in time slot \( t \). The payment function assigns payments based on the output history (independent of agent’s names).

We will discuss in the later subsections how to design this matching and payment function such that it is aligned with the designer’s objective taking into consideration that the users are strategic. Note that although the designer will announce its choice of matching and payment functions to the agents this does not mean that the agents will know at each time who will perform which task and what payments will be received. This is because agents do not know the history of the outputs of the other agents.

**Utility of the agents and the designer:** Since the agents do not observe the outputs of others and hence, have a different observation history than the designer. We separately define the history of observations for each agent. Each agent at the beginning of the time slot is assigned a task and observes the quality of the task. Each agent at the beginning of time slot \( t+1 \) is informed by the designer about how much output it produced in time slot \( t \) and is also paid by the designer for the output produced time slot \( t \). Hence, we can say that an agent in time slot \( t+1 \) will observe the output produced by it in previous time slot, the payment it receives for that output and the task that it is assigned for the time slot \( t+1 \). We write the set of observations of an agent \( i \) up to time \( t \) as \( h_{t}^{i} \). Note that \( h_{0}^{0} = \phi \). We write \( \mathcal{H}_{i}(t) \) to denote the set of all possible private histories up to time \( t \). The set of all the possible histories is \( \mathcal{H}_{i} = \bigcup_{t=0}^{\infty} \mathcal{H}_{i}(t) \). We write the strategy of agent \( i \) that maps the history of observations of the agent to effort level as \( \pi_{i} : \mathcal{H}_{i} \rightarrow [0, e_{i}^{\text{max}}] \)

An agent \( i \) exerts effort \( \pi_{i}(h_{t}^{i}) \) in time slot \( t \) following a private history \( h_{t}^{i} \) and the designer’s observation history \( h_{0}^{i} \) and produces output

\[
W_{i}(h_{t}^{i}, h_{0}^{i}, \pi_{i}|M) = F(i)G(M[h_{0}^{i}, i]) \pi_{i}(h_{t}^{i})
\]

We define the total long-term output\(^4\) over time as follows:

\[
W(\{\pi_{i}\}_{i=1}^{N}|M) = \lim \inf_{T \rightarrow \infty} \frac{1}{T+1} \sum_{t=0}^{T} \sum_{i=1}^{N} W_{i}(h_{t}^{i}, h_{0}^{i}, \pi_{i}|M)
\]

The long-term revenue associated with this output is \( bW(\{\pi_{i}\}_{i=1}^{N}) \). Using the payment function and the cost associated with the effort exerted we can compute

\(^4\)We use liminf to ensure limits always exist.
the utility of the agent $i$ following a private history $h_t^i$ and the designer’s history of $h_0^{t+1}$:

$$u_i(h_0^{t+1}, h_t^i, \pi_i|M, P) = W_i(h_t^i, h_0^t, \pi_i|M)P(h_0^{t+1}, i) - c_i\pi_i(h_t^i)^2$$

Note that the above is the utility for one time slot and we used the notation in such a way that it is clear that agent takes the actions given the knowledge of the matching and the payment function $u_i(h_0^t, h_t^i, \pi_i|M, P)$. We can now define the long-term utility for agent $i$ as follows:

$$U_i(\{\pi_i\}_{i=1}^N|M, P) = \lim \inf_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^T u_i(h_0^t, h_t^i, \pi_i|M, P)$$

Similarly, we can define the utility of the designer (profit) after considering the payments to the agents:

$$u_0(h_0^{t+1}, \{h_t^i, \pi_i\}_{i=1}^N|M, P) = \sum_{i=1}^N [b - P(h_0^{t+1}, i)] W_i(h_0^t, h_t^i, \pi_i|M)$$

We have defined the utility of the designer in one of the time slots $t$. We now define the long-term utility in the same manner as we did before for an agent:

$$U_0(\{\pi_i\}_{i=1}^N|M, P) = \lim \inf_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^T u_0(h_0^{t+1}, \{h_t^i, \pi_i\}_{i=1}^N|M, P)$$

**Objective of the designer and the agents:** In this section we will discuss the objective of the designer and the agents. The designer needs to choose the matching and the payment function to maximize its long-term utility.

$$\max_{M, P} U_0(\{\pi_i\}_{i=1}^N|M, P)$$

Note that the solution to the above maximization in general will depend on the strategy of the agents $\pi_i$ and the quality of the agents $F(i)$, which is not known to the designer. The designer can have a different objective than the above for e.g., the total long-term output.
It is assumed that the agents are rational, know their own utilities, know the matching and the payment function (announced by the designer), but do not know the utility of the other agents, and this is common knowledge among the agents. The designer also does not know the utilities of agents as well. The agents choose strategies such that their utilities are maximized.

$$\max_{\pi_i} U_i(\pi_i, \{\pi_j\}_{j=1,j\neq i}^N|M, P)$$

The solution to the above maximization will in general depend on the strategy of the rest of the agents and the quality of the rest of the agents, which is not known to the agent. If all the agents can maximize the above simultaneously, then it will be a Nash equilibrium (i.e. no agent will want to deviate).

The above maximization problems involve incomplete information for both the designer and the agent. Hence, it is even hard to say that there will exist a solution - matching function, payment function and the strategy of agents that solve both the above optimization problems simultaneously. In fact even if there was a solution to the above problems, then it may still have the following problems. The computation of such a solution may require a separate entity that has access to all the utilities and the limited information each agent conditions its strategy on. Additionally, the complexity of computing such a solution may be too high. In a realistic setting we would like the designer to compute the matching and payment function with the limited information which it possess (locally computable) and the agents should also be able to compute their strategy with limited information they possess.

In our approach we propose a matching and payment function, which we show can achieve high long-term utility $U_0(\{\pi_i\}_{i=1}^N|M, P)$ (Theorem 2), and is locally computable. We will also show that the each agent can maximize $U_i(\pi_i, \{\pi_j\}_{j=1,j\neq i}^N|M, P)$ and thus compute locally its equilibrium strategy.

### 3.1 First impression is the last impression (FILI)

In this section we propose the FILI mechanism. The FILI mechanism has two components:

**A) Payment function:** The payment per unit output for the agent $i$ who pro-
duces $R_i$ units of normalized output is given as $\alpha R_i$. We assume a linear payment function here and the parameter $\alpha$ is a design variable. We will discuss later how the designer should choose $\alpha$. The payment function is made public to all the agents. The payment for the output produced in time slot $t$ is made in time slot $t + 1$ (after observing the normalized output of time slot $t$).

**B) Matching function**

1. **Ranking stage**: This stage corresponds to time slot zero. In this stage, the designer uses a fixed matching rule (chosen randomly) to match the agents with the tasks. Each agent knows the quality of the task upon assignment. Each agent then decides the amount of effort to exert. The output is observed by the designer and the agent itself. The designer computes the normalized output for every agent and ranks them in the increasing order of these normalized outputs. If there are ties in the normalized outputs, then the designer breaks the ties randomly.

2. **Operational stage**: This stage corresponds to time slots starting with time slot one up to infinity. Based on the rankings computed in ranking stage the designer matches the agents to the tasks assortatively. Hence, an agent with rank $k$ is matched to task with rank $k$ (both ranked in increasing order). This matching of the agents to the tasks stays fixed for all the future time slots. Each agent knows the quality of the task upon assignment. Each agent then decides the amount of effort to exert in every time slot.

We write the above mechanism in the form of algorithm given in Algorithm 1. Next we describe the optimal strategies of the agents given the above mechanism.

### 3.2 Threshold based bang-bang strategy

The threshold based bang-bang strategy for agent $i$ is defined as:

1. **Ranking stage**: In time slot zero agent $i$ should exert the maximum effort it can $e^{\text{max}}_i$.

2. **Operational stage**: Suppose agent $i$ is allocated a task with quality $H(i)$ (based on the ranking of the set of the normalized outputs computed in the ranking stage). If $H(i) > \frac{\alpha F(i)}{2}$, then the agent exerts maximum effort in every time slot to follow, otherwise the agent exerts zero effort in every time
Algorithm 1 First impression is the last impression (FILI)

$h_0^0 = \emptyset$

for $t = 0$ to $T-1$ do
  if $t = 0$ then
    Ranking Stage
    Assign tasks based on $M(h_0^t, :)$ (chosen randomly)
    Observe the outputs $W_i(h_0^t, h_i^t, \pi_i)$ for all $i$
    Compute the normalized output for all agents
    $R_i = W_i(h_d^t, h_i^t, \pi_i|M)/(G[M(h_0^t, i)])$
    $\{R_s, I_s\} = \text{Sort}\{R_i\}_{i=1}^N$ where $R_s$ is sorted array of the normalized outputs in the increasing order, $I_s$ is the set of corresponding indices of the elements in the sorted array
  end if
  if $t \geq 1$ then
    Operational Stage
    for $i = 1$ to $N$ do
      $M(h_0^t, I_s(i)) = i$
    end for
    Assign the tasks based on $M(h_0^t, :)$
    Observe the outputs $W_i(h_0^t, h_i^t, \pi_i|M)$ for all $i$
  end if
  $h_0^{t+1} = h_0^t \cup \{W_i(h_0^t, h_i^t, \pi_i)\}_{i=1}^N$
end for

Basically, the above strategy requires the agent to exert the maximum effort in the zeroth time slot and thus secure the highest possible task that can be allocated to it. In the next time slot if it turns out that the task allocated is of sufficiently high quality, then the agent continues to exert maximum effort else, it does not exert any effort. In the next theorem we state the optimality of the proposed strategy.

**Theorem 1:**

- The threshold based bang-bang strategy is the best response of an agent to other agents’ strategies.
- If all the agents follow the threshold based strategy, then it will be a NE
referred to as threshold equilibrium (TE).

- If the cost $c_i$ decreases with quality of the agent and if the maximum effort level $e_{max}^i$ is the same for all the agents, then the utility achieved by every agent in the TE is no less than that in any other NE.

The proofs to all the theorems are in the Appendix section.

### 3.3 Efficiency for the designer

In this section we will show that the proposed mechanism for matching and payment is efficient for the designer as well. In the following proposition we state the performance benchmark for both the long-term output and the long-term utility of the designer and the actual performance achieved by the proposed FILI mechanism in TE. we write the normalized outputs sorted in the increasing order as follows $\{F(m_1)e_{max}^1, ..., F(m_N)e_{max}^N\}$. $I()$ is the indicator function which takes the value 1 when the condition in the argument is true and zero otherwise.

**Proposition 1:**

- The total long-term output is bounded above by $\sum_{i=1}^{N} F(m_i)G(i)e_{max}^i$.
- The total long-term output achieved in the TE is $\sum_{i=1}^{N} F(m_i)G(i)e_{max}^i I(G(i) \geq \frac{c_{m_i}}{\alpha F(m_i)\tau})$.
- The long-term utility of the designer is bounded by $\sum_{i=1}^{N} bF(m_i)G(i)e_{max}^i$.
- The long-term utility of the designer achieved in the TE is $\sum_{i=1}^{N} (b - \alpha F(m_i)e_{max}^i)G(i)e_{max}^i I(G(i) \geq \frac{c_{m_i}}{\alpha F(m_i)\tau})$.

The upper bounds that we prove above are very high performance benchmarks which assume that the agents are not strategic and always exert maximum effort (no moral hazard) and that agent’s qualities are known (no adverse selection). The expressions for the performance of the proposed mechanism depends on the fact that for how many agents find the task to be of sufficiently high quality $G(i) \geq \frac{c_{m_i}}{\alpha F(m_i)\tau}$, which also depends on the design variable $\alpha$. Next we describe how should the designer choose $\alpha$ and then compare the performance of the proposed mechanism with the upper bounds.

In order to have a meaningful comparison with the upper bound we make
some assumptions on the range of the cost, task and agent quality. We denote the upper bound on cost, maximum effort and the task quality for all the agents as $c^{\text{max}}$, $e^{\text{max}}$, $s^{\text{max}}$ and $\epsilon, \delta, \gamma$ are non-negative constants less than one.

**Assumption 1:** For all the agents $i$, $c_i \in [c^{\text{max}}(1 - \epsilon), c^{\text{max}}]$, $e_i^{\text{max}} \in [e^{\text{max}}(1 - \delta), e^{\text{max}}]$. For all tasks $j$, $G(j) \in [s^{\text{max}}(1 - \gamma), s^{\text{max}}]$. Each agent’s quality is drawn independently from a uniform distribution $U \sim [0, q^{\text{max}}]$. In addition $q^{\text{max}} > 1$ and $\frac{b}{c} > \frac{e^{\text{max}}}{s^{\text{max}}(1 - \gamma)}$.

In the above assumption $q^{\text{max}} > 1$ ensures that there is sufficient number of agents with high quality and $\frac{b}{c} > \frac{e^{\text{max}}}{s^{\text{max}}(1 - \gamma)}$ ensures that the benefit to cost ratio is not too low since otherwise incentivizing the agents becomes impossible. Suppose that designer selects $\alpha = \alpha^* = \frac{c}{s^{\text{max}}(1 - \gamma)q^{\text{max}}}$. In order to select this $\alpha$ the designer needs to know the upper bound on the cost, the upper bound on the quality of the agents and lower bound on the quality of the task. In the next theorem we will compare the performance (total long-term output and long-term utility) of the FILI mechanism when the agents play TE with the upper bounds stated in Proposition 1. Note that we will take the expected long-term output and the expected long-term utility, where the expectation will be with respect to the distribution of the quality of the agents. Define $\Theta = (1 - \frac{1}{q^{\text{max}}})(1 - \delta)(1 - \gamma)$.

**Theorem 2:** If the designer uses the FILI mechanism with $\alpha = \alpha^*$ and all agents use the threshold based bang-bang strategy and if the Assumption 1 holds, then

- The ratio of the expected total long-term output and the corresponding upper bound is greater than $\Theta$.
- The ratio of the expected long-term utility for the designer and the upper bound is greater than $(1 - \frac{c^{\text{max}}}{bs^{\text{max}}(1 - \gamma)})\Theta$.

From the above comparison it is clear if $q^{\text{max}}$ is high and the difference in the maximum effort of any two agents is not too large, then $\Theta$ will be high.

In the Appendix we work out an example to show the wide range of scenarios where our approach can do better than the best that can be achieved under conventional random matching rules.

**Computational properties of FILI:** The designer has the main task of ranking the normalized output, which takes $O(N \log N)$ steps. The mechanism is lo-
cally computable, i.e. only requires limited information available to the designer to be implemented. Surprisingly the agents threshold based bang-bang strategy not only is the best response to other agent’s strategies it is computed by an agent only with the limited private information (without knowledge of the qualities of other agents). Next we discuss the second part of the paper.

4 Repeated matching mechanism design in a stochastic setting

In this section we will make the model more realistic by extending it in two dimensions. In practical settings the output of the agent will stochastically depend on the agent’s quality, the effort exerted and the task quality. We assume an additive noise based model to quantify the output produced. Suppose agent \(i\) works on task \(j\) and exerts effort \(e_i\) then the output produced is given as

\[
F(i)G(j)e_i + Z_{ij},
\]

where \(Z_{ij}\) a random variable representing noise. We assume that \(Z_{ij}\) has zero mean and has a finite variance \(\sigma^2_{ij}\). Also, the noise random variables in different time slots are i.i.d.. The noise in the output of the agents are mutually independent random variables. We assume that the mean and the variance of \(Z_{ij}\) is known to the designer. We also relax the assumption that agent’s know their own quality from the first part and allow the agents to learn their qualities based on the output.

In the first part we defined the matching and the payment function in a general manner and they depended on the history of the observations made by the designer. The definition for the matching and the payment function remain the same in this part as well except the history in this case will be stochastic.

Utility of the agents and the designer: Consider that the matching is carried out for a total of \(T + 1\) time slots. The agent’s strategy \(\pi^T_i : \mathcal{H}_i(T + 1) \rightarrow [0, e^\text{max}_i]\) is mapping from histories of length less than \(T + 1\) to effort levels. An agent \(i\) exerts effort \(\pi_i(H^t_i)\) in time slot \(t\) following a private history \(H^t_i\) and the

\[5\text{Note that we use } H^t_i \text{ since the history in this case is a random variable in contrast to } h^t_i \text{ in the first part.}\]
designer’s observation history $H_{0}^{t}$ and produces output

$$W_{i}(H_{i}^{t}, H_{0}^{t}, \pi_{i}^{T} | M) = F(i) G(M [H_{0}^{t}, i]) \pi_{i}^{T}(H_{i}^{t}) + Z_{iM[H_{0}^{t}, i]}$$

The total mean output when there are $T + 1$ time slots in the mechanism is given as,

$$W(\{\pi_{i}^{T}\}_{i=1}^{N} | M) = \frac{1}{T + 1} \sum_{t=0}^{T} \sum_{i=1}^{N} W_{i}(H_{i}^{t}, H_{0}^{t}, \pi_{i} | M)$$

The total long-term output $W(\{\pi_{i}^{T}\}_{i=1}^{N} | M)$ is defined as the almost sure limit of $W(\{\pi_{i}^{T}\}_{i=1}^{N} | M)$ as $T \to \infty$. The utility of the agent $i$ following a private history $H_{i}^{t}$ and the designer’s history of $H_{0}^{t+1}$,

$$u_{i}(H_{0}^{t+1}, H_{i}^{t}, \pi_{i}^{T} | M, P) = W_{i}(H_{i}^{t}, H_{0}^{t}, \pi_{i}^{T} | M) P(H_{0}^{t+1}, i) - c_{i}\pi_{i}(H_{i}^{t})^{2}$$

The long-term utility is defined as the almost sure limit of the utility when there are $T + 1$ slots in the mechanism (in the same way as the total long-term output above). Similarly we can define the utility for the designer as well.

### 4.1 Initial impression is the last impression (IILI)

In this section we propose the IILI mechanism. We will consider that the total number of time slots is $T$ and this information is known to the agents. The IILI mechanism has two components:

**A) Payment function:** We assume that the designer follows a similar payment rule as in the first part. Suppose that the agent $i$ when matched to task $j$ produces an output of $W$. In this case the designer will pay the agent $\alpha \frac{W^{2}}{G(j)} - \alpha \frac{\sigma_{ij}^{2}}{G(j)}$.

If the designer uses the same matching rule as proposed in the first part, then the output and the profits can be very bad because the noise in the output can lead to designer identifying the incorrect agents as high quality ones. We propose a natural extension of the mechanism proposed in the first part. The first stage, i.e. the ranking stage, will now comprise of multiple time slots. In this stage the designer will develop an estimate for the normalized outputs of the agents and then use it in the next stage to match them.

**B) Matching function:**
1. **Ranking stage**: This stage comprises of the first $\sqrt{T} + 1$ time slots\(^6\). In this stage the designer uses any fixed matching rule (chosen randomly) to match the agents with the tasks. Each agent knows the quality of the task upon assignment. Each agent then decides the amount of effort to exert. The output is observed by the designer and the agent. The designer computes the normalized output for every agent $i$ in time slot $t$ as $\tilde{R}^t_i(H^t_i, H^t_0, \pi^T_i | M) = \frac{W_i(H^t_i, H^t_0, \pi^T_i | M)}{G(M[H^t_0, i])}$. At the end of the $\sqrt{T}^{th}$ time slot, which is the end of the ranking stage, the designer computes an estimate of normalized output for every agent $i$ as $\hat{R}\sqrt{T}_i(H^T_i, H^0_T, \pi^T_i | M) = \frac{1}{\sqrt{T}+1} \sum_{t=0}^{\sqrt{T}} \tilde{R}^t_i$. The designer then ranks the agents in the increasing order of their normalized outputs.

2. **Operational Stage**: Based on the ranking of the normalized outputs the designer will match the agents to the tasks. Agent with rank $k$ will be matched to the task with rank $k$. Each agent knows the quality of the task upon assignment. Each agent then decides the amount of effort to exert on the task assigned in every time slot. The task that an agent is matched to remains the same for the time slots to follow.

We write the matching function in the form of algorithm in Algorithm 2.

### 4.2 Threshold based bang-bang learning strategy

The threshold based bang-bang learning strategy for agent $i$ is defined as:

1. **Ranking stage**: In each time slot in the ranking stage agent $i$ should exert the maximum effort $e_i^{max}$. Agent $i$ observes the output in time slot $t$. The agent knows the quality of the task assigned to it and the effort that it exerts, which allows the agent to estimate its quality as follows. Define $\hat{F}(i)^t = \frac{W_i(H^t_i, H^t_0, \pi^T_i | M)}{G(M[H^t_0, i])e_i^{max}}$. The agent at the end of ranking stage (after $\sqrt{T}$ time slots) computes the following estimate for its quality $\hat{F}(i)^{\sqrt{T}} = \frac{1}{\sqrt{T}+1} \sum_{t=0}^{\sqrt{T}} \hat{F}(i)^t$.

2. **Operational stage**: Agent $i$ is allocated the task based on the ranking of the mean normalized outputs calculated at the end of stage one, which we denote as $H'(i)$. If $H'(i) > \frac{c_i}{\alpha(F(i)^{\sqrt{T}})^2}$, then the agent exerts the effort $e_i^{max}$ else, the agent exerts zero effort.

---

\(^6\)We assume that $\sqrt{T}$ is an integer for convenience.
Next, we will show that the long-term utility achieved by the above strategy is optimal. Since the long-term utility of the agent is a random variable, following the proposed strategy is a best response iff it leads to incentive compatibility across all the possible random realizations (except a set of probability measure zero).

**Theorem 3**

- The threshold based bang-bang learning for an agent is the optimal best response to other agents’ strategies.
- If all the agents follow the threshold based bang-bang learning strategy, then it comprises a NE referred to as threshold learning equilibrium (TLE).
- If the cost $c_i$ decreases with the quality of the agent and if the maximum effort level $e_i^{\text{max}}$ is the same for all the agents, then the utility achieved by every agent in the TLE is no less than the utility in any other NE.

The above optimality result is interesting because the proposed learning procedure is the NE for the foresighted agents who at the beginning of the ranking stage are not even aware of their own quality.

### 4.3 Efficiency for the designer

In this section we will prove the efficiency of the proposed method in terms of the long-term output and the long-term utility of the designer.

**Theorem 4:**

- The long-term output achieved by the IILI mechanism in the TLE is the same as the total output of the FILI mechanism in TE with probability one.
- The long-term utility of the designer achieved by IILI mechanism in the TLE is the same as the long-term utility of the FILI mechanism in TE with probability one.

The above theorem shows that the designer is able to ensure as much profit as it was able to get when the output was not stochastic. Note that the upper bound on the long-term utility and the long-term output output derived in Proposition 1 continue to hold even for the above case with stochastic outputs. This means that the result from Theorem 2 continues to hold even in this case.
Algorithm 2 Initial impression is the last impression

\[ H_0^0 = \phi \]

for \( t = 0 \) to \( T - 1 \) do

if \( t \leq \sqrt{T} \) then

Ranking Stage

Assign tasks based on \( M(H_0^0, : ) \)

Observe the outputs

\[ H_{t+1}^t = H_0^t \cup \{ W_i(H_d^t, H_i^t, \pi_i^T | M) \}_{i=1}^N \]

Compute the normalized output for all agents

\[ \bar{R}_i^t(H_i^t, H_0^t, \pi_i^T | M) = \frac{W_i(H_d^t, H_i^t, \pi_i^T | M)}{\{(G[M(H_0^t, i)])}} \]

end if

for \( i = 1 \) to \( N \) do

\[ \bar{R}_{i}^{\sqrt{T}}(H_i^{\sqrt{T}}, H_0^{\sqrt{T}}, \pi_i^T | M) = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{\sqrt{T}} \bar{R}_i^t \]

end for

\[ \{ R_s, I_s \} = \text{Sort}(\{ \bar{R}_i^{\sqrt{T}} \}_{i=1}^N) \] where \( R_s \) is sorted array of the normalized outputs in the increasing order, \( I_s \) is the set of corresponding indices of the elements in the sorted array

if \( t > \sqrt{T} \) then

Operational Stage

for \( i = 1 \) to \( N \) do

\[ M(H_0^t, I_s(i)) = i \]

end for

Assign tasks based on \( M(H_0^t, : , t) \)

end if

end for

In the next section we define the notion of regret for the agents and compute it to show that the proposed learning procedure are efficient in terms of rate of learning.

Regret for the agents: In the previous sections we showed that the threshold based bang-bang learning strategy converges to the optimum as \( T \to \infty \). In this section we will discuss the efficiency of the described learning approach when there is a finite number of time slots \( T \) in the mechanism. Next we define the notion of regret. We write the expected utility that an agent can achieve following our proposed threshold based learning strategy, where the expectation is taken with respect to the joint distribution of the noise random variable in the output,
as $\bar{U}_i(T)$. We will compare it to the maximum expected utility denoted as $\bar{U}_i^*(T)$ that can be achieved by an agent who knows his quality perfectly. The regret is defined as the difference between the two $Reg_i(T) = \bar{U}_i^*(T) - \bar{U}_i(T)$. In the next theorem we will show that the above regret is sublinear. We define a constant $\zeta_i$ for agent $i$ (See Appendix for the expression).

**Theorem 5**: The regret for every agent $i$ decreases as $\sqrt{T}$.

The above theorem shows that the rate of learning is fast for the agents. Next we state the conclusions.

## 5 Conclusion

In this work we developed a mechanism to learn how to match and incentivize the agents efficiently and thus simultaneously mitigate the problems of adverse selection and moral hazard. The model considered only requires the designer to act on the outputs (stochastic) produced by the strategic agents and also works in settings where the agents may not know their own quality and must learn it. We show that the proposed mechanism ensures that the designer learns how to mitigate both moral hazard and adverse selection thereby achieving high profits. We also show that given the proposed mechanism, the agents find in their self interest to follow the simple threshold based bang-bang strategies that allow them to learn their own qualities and achieve the highest possible utilities among all NEs.
Appendix

In all the proofs we will use $I(A)$ as the indicator function. If the condition $A$ holds, then the indicator is one and zero otherwise.

Theorem 1:

1. The threshold based bang-bang strategy is the best response of an agent to other agents’ strategies.
2. If all the agents follow the threshold based strategy above, then it will be a NE referred to as threshold equilibrium (TE).
3. If the cost $c_i$ decreases with quality of the agent and if the maximum effort level $e_{i,max}$ is the same for all the agents, then the utility achieved by every agent in the TE is no less than that in any other NE.

Proof: We will start out by writing the long-term utility of an agent for the FILI mechanism as follows. We write the agent $i$’s effort in time slot $t$ as $e^t_i$, which has to be equal to $\pi_i(h^t_i)$. We denote the task allocated to the agent after ranking the set of normalized outputs $\{F(j)e^0_j\}_{j=1}^N$ as $D(i, \{F(j)e^0_j\}_{j=1}^N)$

$$U_i(\{\pi_j\}_{j=1}^N|M, P) = \lim_{T \to \infty} \frac{1}{T+1} \sum_{t=1}^T \alpha F(i)^2 D(i, \{F(j)e^0_j\}_{j=1}^N)(e^t_i)^2 - c_i(e^t_i)^2$$

$$= \lim_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^T (e^t_i)^2 \left[\alpha F(i)^2 D(i, \{F(j)e^0_j\}_{j=1}^N) - c_i\right]$$

We denote $\lim_{T \to \infty} \sum_{t=0}^T \frac{1}{T+1}(e^t_i)^2 = \bar{e}^2_i$. Then we can write the above utility function as

$$U_i(\{\pi_j\}_{j=1}^N|M, P) = \bar{e}^2_i \left[\alpha F(i)^2 D(i, \{F(j)e^0_j\}_{j=1}^N) - c_i\right]$$

Next we want to solve the following.

$$\max_{\pi_i} U_i(\{\pi_j\}_{j=1}^N|M, P)$$
Note that \( \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1}^N) - c_i \) is an increasing function for agent \( i \)'s effort in the zeroth time slot. Therefore, the following is true:

\[
U_i(\{ \pi_j \}_{j=1}^N | M, P) = e_i^2 \left[ \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1}^N) - c_i \right] \\
= e_i^2 \left[ \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1,j\neq i}^N, F(i)e_i^{\max}) - c_i \right]
\]

The above inequality basically shows the maximum value that can be achieved by only choosing the effort of agent \( i \) in the zeroth time slot \( e_i^0 \).

Now suppose

\[
\left[ \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1,j\neq i}^N, F(i)e_i^{\max}) - c_i \right] > 0
\]

in this case it is clear that \( e_i^2 \rightleftharpoons (e_i^{\max})^2 \) leads to the maximum of the above expression.

Hence, we can write

\[
U_i(\{ \pi_j \}_{j=1}^N | M, P) \leq (e_i^{\max})^2 \left[ \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1,j\neq i}^N, F(i)e_i^{\max}) - c_i \right] \forall \pi_i
\]

Now suppose

\[
\left[ \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1,j\neq i}^N, F(i)e_i^{\max}) - c_i \right] \leq 0
\]

in this case it is clear that \( e_i^2 = 0 \) leads to the maximum of the above expression. Hence, the maximum of the utility in this case is given as zero. We can combine the above two to get the maximum value

\[
\max_{\pi_i} U_i(\{ \pi_j \}_{j=1}^N | M, P) = \\
(\max_{e_i} e_i^{\max})^2 \left[ \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1,j\neq i}^N, F(i)e_i^{\max}) - c_i \right] \times \\
I(\alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1,j\neq i}^N, F(i)e_i^{\max}) \geq c_i)
\]

Now we see the utility achieved by our proposed strategy for agent \( i \). Our strategy will mean that agent \( i \) exerts maximum effort in the zeroth time slot and is allocated to \( D(i, \{ F(j)e_j^0 \}_{j=1}^N) \). In the next time slots it will do either of the following.
• If $F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1, j \neq i}^N, F(i)e_i^{\text{max}}) \geq \frac{c_i}{\alpha}$, then the agent will exert maximum effort in all the following periods and the utility is given as $(e_i^{\text{max}})^2 \left[ \alpha F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1, j \neq i}^N, F(i)e_i^{\text{max}}) - c_i \right]$

• If $F(i)^2 D(i, \{ F(j)e_j^0 \}_{j=1, j \neq i}^N, F(i)e_i^{\text{max}}) < \frac{c_i}{\alpha}$, then the agent will exert zero effort in all the following periods and the utility is given as $0$.

Hence, we can see that the proposed strategy can achieve the maximum value we showed above. Therefore, the proposed strategy is the optimal given other agents follow any strategy or we can say that the proposed strategy is the best response to the other agents' strategies. This proves part 1. Part 2 is obvious because if all the agents best respond then the resulting joint strategy is NE. For part 3 we refer to the proof of Theorem 3, which proves the more general version of the statement here.

\[ \square \]

We write the set of normalized outputs as follows $\{F(1)e_1^{\text{max}}, ..., F(N)e_N^{\text{max}}\}$ and we write the normalized outputs sorted in the increasing order as follows $\{F(m_1)e_{m_1}^{\text{max}}, ..., F(m_N)e_{m_N}^{\text{max}}\}$

**Proposition 1:**

• The total long-term output is bounded above by $\sum_{i=1}^N F(m_i)G(i)e_{m_i}^{\text{max}}$

• The total long-term output achieved in the TE is $\sum_{i=1}^N F(m_i)G(i)e_{m_i}^{\text{max}}I(F(m_i)^2G(i) \geq \frac{c_i}{\alpha})$

• The long-term utility of the designer is bounded by $\sum_{i=1}^N bF(m_i)G(i)e_{m_i}^{\text{max}}$

• The long-term utility of the designer achieved in the TE is $\sum_{i=1}^N (b - \alpha F(m_i)e_{m_i}^{\text{max}})G(i)e_{m_i}^{\text{max}}I(F(m_i)^2G(i) \geq \frac{c_i}{\alpha})$

**Proof.** Let us first establish the upper bound on the profit and the output. We know that the long-term profit for the firm is given as

\[ \text{Profit} = \sum_{i=1}^N F(m_i)G(i)e_{m_i}^{\text{max}} \]
\[ U_0(\{\pi_i\}_{i=1}^N|M, P) = \lim \inf_{T \to \infty} \frac{1}{T+1} \sum_{t=0}^{T} u_0(h_{d}^{t+1}, \{h_{i}^{t}\}_{i=1}^N, \{\pi_i\}_{i=1}^N|M, P) \]

We will find the upper bound on \( u_0(h_{d}^{t+1}, \{h_{i}^{t}\}_{i=1}^N, \{\pi_i\}_{i=1}^N|M, P) \) and then use it to compute the upper bound on the long-term profit.

\[ u_0(h_{d}^{t+1}, \{h_{i}^{t}\}_{i=1}^N, \{\pi_i\}_{i=1}^N|M, P) \leq b \sum_{i=1}^{N} W_i(h_{d}^{t}, h_{i}^{t}, \pi_i|M) \]

We now compute the upper bound on the total output that is possible in one period. Since we are computing upper bound we will assume that the effort level is decided in order to maximize the output and not necessarily the agent’s utilities. For any matching to have the maximum output each agent \( i \) should exert maximum effort \( e_{\text{max}}^i \) otherwise the effort can always be increased to improve the output. Hence, in this case the problem reduces to finding the optimal matching of agents exerting maximum efforts to the tasks. Consider a general matching \( M' : P \to T \). We write the set of normalized outputs as follows \( \{F(1)e_{\text{max}}^{\pi_1}, ..., F(N)e_{\text{max}}^{\pi_N}\} \) and we write the normalized outputs sorted in the increasing order as follows \( \{F(m_1)e_{\text{max}}^{\pi_{m_1}}, ..., F(m_N)e_{\text{max}}^{\pi_{m_N}}\} \)

We can write the output for this case as follows

\[ \sum_{i=1}^{N} F(i)e_{\text{max}}^i G(M'(i)) \leq \sum_{i=1}^{N} F(m_i)e_{\text{max}}^i G(i), \forall M' \]

The above inequality follows from rearrangement inequality. Hence, the following is obvious too.

\[ \sum_{i=1}^{N} W_i(h_{d}^{t}, h_{i}^{t}, \pi_i) \leq \sum_{i=1}^{N} F(m_i)e_{\text{max}}^i G(i), \forall M' \]

Therefore,

\[ u_0(h_{d}^{t}, \{h_{i}^{t}\}_{i=1}^N, \{\pi_i\}_{i=1}^N|M, P) \leq b \sum_{i=1}^{N} F(m_i)e_{\text{max}}^i G(i) \]
which implies the long-term profit is also bounded above by the same term. Since 
\[ \sum_{i=1}^{N} W_i(h^t_i, h^i, \pi_i) \leq \sum_{i=1}^{N} F(\theta_i)e_{m_i}^{\max} G(i) \] the result for the total long-term output also follows.

Now we compute the profit achieved by the proposed algorithm provided all
the agents follow the threshold based strategy.

We know that an agent \( i \) will exert \( e_{i}^{\max} \) iff \( F(i)2G(M\{e_{j}^{\max}\}_{j=1}^{N}, i) \geq \frac{c_i}{\alpha} \). The contribution of this agent to the designer’s profit is given as follows.

\[
(b - \alpha F(i)e_{i}^{\max})F(i)e_{i}^{\max} G(M\{F(j)e_{j}^{\max}\}_{j=1}^{N}, i)I(F(i)2G(M\{e_{j}^{\max}\}_{j=1}^{N}, i) \geq \frac{c_i}{\alpha})
\]

Therefore, the total profit for the designer can be equivalently written as

\[
\sum_{j=1}^{N} (b - \alpha F(m_j)e_{m_j}^{\max})F(\theta_j)e_{m_j}^{\max} G(j)I(F(m_j)^2G(j) \geq \frac{c_j}{\alpha})
\]

Note that since the profit is the same in each period starting time slot one the long-term profit is the same as the above value. Similarly, the total output can be written as follows.

\[
\sum_{i=1}^{N} F(m_j)e_{m_j}^{\max} G(j)I(F(m_j)^2G(j) \geq \frac{c_j}{\alpha})
\]

\[
\]

**Proof.** Note that \( c \) is the upper bound on the cost for any agent. Therefore, if there exists a \( k \) for which \( F(m_k)^2G(k) \geq \frac{c}{\alpha} \), then all the agents \( j \geq k \) will exert maximum effort. Substitute \( \alpha = \frac{c}{\delta_{\min}q_{\max}} \) and we get the following condition

\[
F(m_k)^2 \geq \frac{e_{\max}(1-\gamma)q_{\max}}{G(k)}.
\]

Hence, it is sufficient that if an agent has a quality greater than \( \sqrt{q_{\max}} \), then the agent should exert maximum effort. We can compute the lower bound on the output of the proposed algorithm as follows.
\[
E \left[ \sum_{i=1}^{N} F(m_i)G(i) e_{m_i}^{\text{max}} I(F(m_i)^2 G(i)) \right] \geq \frac{e_i}{\alpha} \]
\[
e^{\text{max}} s^{\text{max}} (1 - \delta)(1 - \gamma) E \left[ \sum_{i=1}^{N} F(m_i) I(F(m_i) \geq \sqrt{q^{\text{max}}}) \right]
\]
\[
= e^{\text{max}} s^{\text{max}} (1 - \delta)(1 - \gamma) \sum_{i=1}^{N} E \left[ F(i) I(F(i) \geq \sqrt{q^{\text{max}}}) \right]
\]

We now compute
\[
E[F(i)I(F(i) \geq \sqrt{q^{\text{max}}})] = \frac{1}{q^{\text{max}}} \int_{\sqrt{q^{\text{max}}}}^{q^{\text{max}}} x \, dx
\]
\[
= \frac{q^{\text{max}} - 1}{2}
\]

Therefore,
\[
E \left[ \sum_{i=1}^{N} F(m_i)G(i) e_{m_i}^{\text{max}} I(F(m_i)^2 G(i)) \right] \geq e^{\text{max}} s^{\text{max}} (1 - \delta)(1 - \gamma) N \frac{q^{\text{max}} - 1}{2}
\]

Based on the same method we can get the upper bound long-run output
\[
N \frac{q^{\text{max}}}{2} s^{\text{max}} e^{\text{max}}
\]

We can take the ratio of the two expressions above and get the final result as follows.
\[
\Theta = (1 - \delta)(1 - \gamma)(1 - \frac{1}{q^{\text{max}}})
\]

Next we will show the bounds for the profit achieved by the designer.
Similar to the approach described for the ratio of output we can obtain the ratio of the profits as well, which is given below.

\[
(1 - \frac{c_i e_i^{\text{max}}}{b_i^{\text{max}} (1 - \gamma)})(1 - \frac{1}{q_i^{\text{max}}})(1 - \delta)(1 - \gamma)
\]

\[\square\]

**Theorem 3**

- The threshold based bang-bang learning for an agent is the optimal best response to other agents’ strategies.

- If all the agents follow the threshold based bang-bang learning strategy, then it comprises a NE referred to as threshold learning equilibrium (TLE).

- If the cost \( c_i \) decreases with the quality of the agent, then the utility achieved by every agent in the TLE (i.e., every agent follows the threshold based learning strategy) is no less than the utility in any other NE.

**Proof.** The designer will use IILI to match and pay the agents. We write the strategy for agent \( i \) when the matching occurs for a total of \( T \) stages as a mapping from the history of its private observations \( h_i^T \) and to the effort levels \( \pi_i^T : H_i^T \to [0, e_i^{\text{max}}] \), where \( H_i^T \) is the set of all the possible observations up to time \( T \). Note that the history of the agents and the designer in this case are random variables because the output produced is stochastic. We will define the noise random variables and the other random variables to be used in the problem over the following probability space \( \{\Omega, \mathcal{F}, P\} \) with \( \Omega \) as the sample space, \( \mathcal{F} \) as the sigma field of events and \( P \) as the probability measure. Noise in the output at time \( t \) for agent \( i \) working on task \( j \) is given as the random variable \( Z_{ij}^t : \Omega \to \mathbb{R} \), whose mean is zero and variance is \( \sigma_{ij}^2 \). The random variables \( \{Z_{ij}^t, i \in \{1, ..., N\}, j \in \{1, ..., N\} \text{ and } t \in \{0, \ldots, \infty\}\} \) are mutually independent. The random variables across time \( Z_{ij}^t \) and \( Z_{ij}^{t'} \) have identical distribution as well.

We define the histories for the agent and the designer as follows. For agent \( i \) at time \( t \) \( H_i^t \) is the random variable that contains the set of observations up to time \( t \). For the designer at time \( t \) \( H_0^t \) is the random variable that contains the set of
observations up to time \( t \). We initialize \( H^0_i = \phi \) and \( H^0_0 = \phi \). The agent observes the normalized output

\[
\tilde{R}_i(H^t_i, H^t_0, \pi^T_i | M) = F(i)\pi^T_i(H^t_i) + Z_{iM(H^t_0, i)} G(M[H^t_0, i])
\]

The task quality of the task assigned to agent \( i \) in time slot \( t \) given as \( G(M[H^t_0, i]) \).

We denote a realization of random variable normalized output when \( \omega \) is the outcome as.

\[
\tilde{R}_i(H^t_i(\omega), H^t_0(\omega), \pi^T_i | M) = F(i)\pi^T_i(H^t_i(\omega)) + Z_{iM(H^t_0(\omega), i)} G(M[H^t_0(\omega), i])
\]

Hence, the history for agent \( i \) in time slot \( t + 1 \) is given as \( H^{t+1}_i = H^t_i \cup \{\tilde{R}_i, G(M[H^t_0, i])\} \). The designer observes the normalized output for all the agents and its history is given as \( H^{t+1}_0 = H^t_0 \cup \{\tilde{R}_j\}_{j=1}^N \). Having defined the strategies and the histories, we can define the utility of the agent for the \( T \) stage mechanism.

Let us first distinguish how does the matching rule work in the two stages. In the ranking stage the matching function chooses a random task for the agents. Hence, the matching for the first \( \sqrt{T} \) time slots is a random variable defined over the same probability space given above. We call the matching for the ranking stage as \( M^{\text{rank}} \in F^* \), where \( F^* \) is the space of bijective mappings from \( \mathcal{P} \to S \).

We refer to \( M^{\text{rank}}(\omega) \) as the realization of the matching and \( M^{\text{rank}}(\omega, i) \) denotes the task assigned to agent \( i \) by the matching. The normalized output in the ranking stage is given as

\[
\tilde{R}_i(H^t_i; H^t_0, \pi^T_i | M^{\text{rank}}) = F(i)\pi^T_i(H^t_i) + Z_{iM^{\text{rank}}(\omega, i)} G(M^{\text{rank}}(\omega, i))
\]

In the above expression \( M^{\text{rank}}(\omega, i) \) denotes the random variable that maps to the task that the agent \( i \) will be assigned to in the ranking stage. Note that the above
expression in RHS does not depend on $H^t_0$ because the task assignment is determined randomly for the ranking stage. Hence, we can call the normalized output in the ranking stage as $\tilde{R}_i(H^t_i, \pi^T_i | M^{rank}(:, i))$. We denote the actual output as

$$W_i(H^t_i, \pi^T_i | M^{rank}(:, i)) = \tilde{R}_i(H^t_i, \pi^T_i | M^{rank}(:, i))G(M^{rank}(:, i))$$

Also, to show consistency we state that $M(H^t_0, i) = M^{rank}(i, i)$ for all $i$ and $t \leq \sqrt{T}$.

In the operational stage, the designer will compare the estimate of normalized outputs.

$$\hat{R}^{\sqrt{T}}(H^t_i, \pi^T_i | M^{rank}) = \frac{1}{\sqrt{T} + 1} \sum_{t=0}^{\sqrt{T}} \hat{R}_i(H^t_i, \pi^T_i | M^{rank})$$

The matching rule for the operational stage is $M^{opnl}$ and the matching for agent $i$ is denoted as $M^{opnl}(\{\hat{R}^{\sqrt{T}}(H^t_i, \pi^T_i | M^{rank})\}_{i=1}^N, i)$. $M^{opnl}$ matches the agents to the tasks assortatively based on the estimate of normalized outputs and the ties will be broken randomly. The normalized output in the ranking stage is given as

$$Z_{iM^{opnl}(\{\hat{R}^{\sqrt{T}}(H^t_i, \pi^T_i | M^{rank})\}_{i=1}^N, i))} = \frac{1}{\sqrt{T} + 1} \sum_{t=0}^{\sqrt{T}} \hat{R}_i(H^t_i, \pi^T_i | M^{rank})$$

We denote the actual output as

$$W_i(H^t_i, H^t_0, \pi^T_i | M^{opnl}) = \hat{R}_i(H^t_i, H^t_0, \pi^T_i | M)G(M^{opnl}(\{\hat{R}^{\sqrt{T}}(H^t_i, \pi^T_i | M^{rank})\}_{i=1}^N, i))$$

For the consistency of notation we state that $M(H^t_0, i) = M^{opnl}(\{\hat{R}^{\sqrt{T}}(H^t_i, \pi^T_i | M^{rank})\}_{i=1}^N, i)$ for all $i$ and $t \geq \sqrt{T}$.

We next write the utility for agent $i$ when there are a total of $T + 1$ slots in the mechanism.
\[ U^T_i \left( \{ \pi_j^T \}_{j=1}^N \mid M, P \right) = U^{\text{rank}, T}_i \left( \{ \pi_j^T \}_{j=1}^N \mid M^{\text{rank}}, P \right) + U^{\text{opnl}, T}_i \left( \{ \pi_j^T \}_{j=1}^N \mid M^{\text{opnl}}, P \right) \]

\[ U^{\text{rank}, T}_i \left( \{ \pi_j^T \}_{j=1}^N \mid M^{\text{rank}}, P \right) = \frac{1}{T+1} \left( \sum_{t=0}^{\sqrt{T}} \left( \alpha W_i(H_i^t, H_0^t, \pi_i^T| M^{\text{rank}}) \right)^2 \cdot \frac{G[M^{\text{rank}}(\cdot, i)]}{G[M^{\text{rank}}(\cdot, i)]} - \alpha \frac{\sigma_i^2 M^{\text{rank}}(\cdot, i)}{G[M^{\text{rank}}(\cdot, i)]} + c_i \pi_i(H_i^t)^2 \right) \]

Note that in the above expression for \( U^{\text{rank}, T}_i \left( \{ \pi_j^T \}_{j=1}^N \mid M^{\text{rank}}, P \right) \) the RHS does not depend on the actions of other agents. We will denote the realization of the utility for ranking stage as \( U^{\text{rank}, T}_i \left( \{ \pi_j^T \}_{j=1}^N, \omega \mid M^{\text{rank}}, P \right) \).

Similarly we can write the expression for the utility of the operational stage.

\[ U^{\text{opnl}, T}_i \left( \{ \pi_j^T \}_{j=1}^N \mid M^{\text{opnl}}, P \right) = \frac{1}{T+1} \left( \sum_{t=\sqrt{T}}^{T} \left( \alpha W_i(H_i^t, H_0^t, \pi_i^T| M^{\text{opnl}}) \right)^2 \cdot \frac{G[M^{\text{opnl}}(\{ \hat{R}_i^T(H_i^T, \pi_i^T|M^{\text{rank}}) \}_{i=1}^N, i))]}{G[M^{\text{opnl}}(\{ \hat{R}_i^T(H_i^T, \pi_i^T|M^{\text{rank}}) \}_{i=1}^N, i))} - \alpha \frac{\sigma_i^2 \hat{M}(\{ \hat{R}_i^T(H_i^T, \pi_i^T|M^{\text{rank}}) \}_{i=1}^N, i))}{G[M^{\text{opnl}}(\{ \hat{R}_i^T(H_i^T, \pi_i^T|M^{\text{rank}}) \}_{i=1}^N, i))} + c_i \pi_i(H_i^t)^2 \right) \]

We will denote the realization of the utility for ranking stage as \( U^{\text{opnl}, T}_i \left( \{ \pi_j^T \}_{j=1}^N, \omega \mid M^{\text{opnl}}, P \right) \).

Hence, the total utility is denoted as

\[ U^T_i \left( \{ \pi_j^T \}_{j=1}^N, \omega \mid M, P \right) = U^{\text{rank}, T}_i \left( \{ \pi_j^T \}_{j=1}^N, \omega \mid M^{\text{rank}}, P \right) + U^{\text{opnl}, T}_i \left( \{ \pi_j^T \}_{j=1}^N, \omega \mid M^{\text{opnl}}, P \right) \]

We will now compute the limit of the above expression as \( T \to \infty \), i.e.,

\[ \lim_{T \to \infty} U^T_i \left( \{ \pi_j^T \}_{j=1}^N, \omega \mid M, P \right) \]. We will first compute the limit of the operational part.

Let us first compute the following expression, which is the first term from the summation for the operational part given above.
\[
\frac{1}{T + 1} \sum_{t=\sqrt{T}+1}^{T} \left( W_i(H_i'(\omega), H_0'(\omega), \pi_i^T | M_{\text{opnl}}^2) \right) = \\
\frac{1}{T + 1} \sum_{t=\sqrt{T}+1}^{T} \left( \left[ F(i)\pi_i^T(H_i'(\omega))G(M_{\text{opnl}}^2(\{\hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T\}_i=1, i)) \right]^2 + 2 \left[ F(i)\pi_i^T(H_i'(\omega))G(M_{\text{opnl}}^2(\{\hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T\}_i=1, i)) \right] \right) \\
+ (Z_{ij}^T(\omega))^2)
\]

The limit of the first term in the above is written as follows.
\[
F(i)^2 \lim_{T \to \infty} G(M_{\text{opnl}}^2(\{\hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T\}_i=1, i)) \right)^2 \lim_{T \to \infty} \frac{1}{T + 1} \sum_{t=\sqrt{T}+1}^{T} \left[ \pi_i^T(H_i'(\omega)) \right]^2
\]

We write \( \lim_{T \to \infty} \frac{1}{\sqrt{T} + 1} \sum_{t=0}^{T} \left[ \pi_i^T(H_i'(\omega)) \right] = \bar{\Pi}_{\text{opnl}}^2(\omega) \) (we assume here that the limit exists).
\[
\hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T) = \\
\frac{1}{\sqrt{T} + 1} \sum_{t=0}^{\sqrt{T}} \left[ F(i)\pi_i^T(H_i'(\omega)) + \frac{Z_{iM_{\text{opnl}}^2(\{\hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T\}_i=1, i))}{G(M_{\text{opnl}}^2(\{\hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T\}_i=1, i))} \right)
\]

We write \( \lim_{T \to \infty} \frac{1}{\sqrt{T} + 1} \sum_{t=0}^{\sqrt{T}} \pi_i^T(H_i'(\omega)) = \bar{\Pi}_{\text{rank}}^T(\omega) \) (we assume here that the limit exists). The second term in the above summation will converge to zero over a set of \( \omega \) with probability one. Therefore,
\[
P(\omega : \lim_{T \to \infty} \hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T) = F(i)\bar{\Pi}_{\text{rank}}^T(\omega)) = 1
\]

Hence, \( \hat{R}_i^{\sqrt{T}}(H_i'(\omega), \pi_i^T | M_{\text{rank}}^T) \to F(i)\bar{\Pi}_{\text{rank}}^T(\omega) \) almost surely. We need to compute the \( \lim_{T \to \infty} G(M_{\text{opnl}}^2(\{\hat{R}_i^{\sqrt{T}}(H_j'(\omega), \pi_i^T | M_{\text{rank}}^T\}_i=1, i)) \). Suppose all the values in \( \{F(j)\bar{\Pi}_{\text{rank}}^T(\omega)\}_j=1 \) are distinct, which means there exists at least a distance \( \zeta > 0 \) between these values.
We denote the matching when these mean normalized outputs are matched assortatively as $M^{\text{opnl}}(\{F(j)\bar{\Pi}^{\text{rank}}_j(\omega)\})_{j=1}^N$. Here $M^{\text{opnl}}(\{F(j)\bar{\Pi}^{\text{rank}}_j(\omega)\})_{j=1}^N$ sorts the agents according to the values of $\{F(j)\bar{\Pi}^{\text{rank}}_j(\omega)\}$ and matches them assortatively to the tasks in all the future time slots, i.e., the operational stage.

Since $P(\omega : \lim_{T \to \infty} \hat{R}^T_i(H_i^T(\omega), \pi^T_i|^{\text{rank}}_i) = F(i)\bar{\Pi}^{\text{rank}}_i(\omega)) = 1$. Therefore, for sufficiently large $T$, each $\hat{R}^T_j(H_j^T(\omega), \pi^T_j|^{\text{rank}}_j)$ will get sufficiently close to $F_j^{\text{rank}}(\omega)$ and thus we get

$$\lim_{T \to \infty} G(M^{\text{opnl}}(\{\hat{R}^T_i(H_i^T(\omega), \pi^T_i|^{\text{rank}}_i\})_{i=1}^N, i))) = G(M^{\text{opnl}}(\{F(j)\bar{\Pi}^{\text{rank}}_j(\omega)\})_{j=1}^N, i))$$

It may so happen that values in this $\{F(j)\bar{\Pi}^{\text{rank}}_j(\omega)\}_{j=1}^N$ are not distinct. So for the agents who have the same mean normalized outputs in the ranking stage the ties will be broken randomly. In such a case one agent may some times get a lower task despite the same mean normalized output. However, in this case the value of the utility if the limit exists will always (with probability one) be between the highest and the lowest tasks that can be achieved in the case of a tie.

We next evaluate the second term in the summation.

$$\frac{1}{T+1} F(i) G(M^{\text{opnl}}(\{\hat{R}^T_i(H_i^T(\omega), \pi^T_i|^{\text{rank}}_i\})_{i=1}^N, i))) \sum_{t=\sqrt{T}}^T \left[ \pi^T_t(H_i^t(\omega)) \times Z_{iM^{\text{opnl}}(\{\hat{R}^T_i(H_i^T(\omega), \pi^T_i|^{\text{rank}}_i\})_{i=1}^N, i))) (\omega) \right]$$

It is easy to see that the expected value of the above summation is zero. The reason is that the noise random variable is independent of the effort strategy. Let us write $M^{\text{opnl}}(\{\hat{R}^T_i(H_i^T(\omega), \pi^T_i|^{\text{rank}}_i\}) = M^{\text{opnl}}(i)$ for concise representation. Next let us compute $\frac{1}{T+1} \sum_{t=\sqrt{T}+1}^T \pi^T_t(H_i^t(\omega)) Z_{iM^{\text{opnl}}(i)))$. Note that $\pi^T_t(H_i^t(\omega))$ is the agent’s decision at time $t$ and it depends on the history of outputs observed, the payments made to the agent and the tasks assigned to the agent, which is given as $h_i^t$. We write the expected value of the above summation as

$$\frac{1}{T+1} \sum_{t=\sqrt{T}+1}^T E[\pi^T_t(H_i^t) Z_{iM^{\text{opnl}}(i))]}$$
Since $Z^t_{iM^{opnl}(i)}$ is independent of the random variables observed before time slot $t$ the expected value above is zero. Also, the variance of the above term can also be derived as follows.

$$E \left[ \left( \frac{1}{T + 1} \sum_{t=\sqrt{T+1}}^{T} \pi^T_i (H^t_i) Z^t_{iM^{opnl}(i)} \right)^2 \right]$$

There are two types of terms when the above summation is expanded. First type are

$$E[(\pi^T_i (H^t_i) Z^t_{iM^{opnl}(i)})^2] = E[\pi^T_i (H^t_i)]^2 E[Z^t_{iM^{opnl}(i)}] \leq \sigma^2_{iM^{opnl}(i)} (e_{i}^{max})^2$$

The terms of the second type are defined next. Suppose we consider two time indices $t$ and $\tau$, where $\tau > t$.

$$E[(\pi^T_i (H^t_i) Z^t_{iM^{opnl}(i)})(\pi^T_i (H^\tau_i) Z^\tau_{iM^{opnl}(i)})] = E[(\pi^T_i (H^t_i) Z^t_{iM^{opnl}(i)}) (\pi^T_i (H^\tau_i) E[Z^\tau_{iM^{opnl}(i)}] = 0$$

Hence, we can see that the variance of the above term will be bounded above by $\frac{1}{T + 1} (e_{i}^{max})^2 (\sigma^2_{iM^{opnl}(i)})^2$. Hence, by using Chebyshev’s inequality we can get the convergence in probability for the limit of the above term. We can extend this to almost sure convergence (using central limit theorem) to approximate the distribution of the random variables above by a normal variable and then use Borel-Cantelli lemma. The approach is similar to the one described above for the first term in the summation, which is why do not mention it here again.

The third term clearly converges to $\sigma^2_{iM^{opnl}(i)}$ almost surely (Strong law of large numbers).

Hence, the expression for the utility summing the ranking and the operational stage is given as (with probability one) for the case when there are no ties in the mean normalized outputs is.

$$(\alpha F(i)2G(M^{opnl}(\{F(j)\Pi_j^{rank}(\omega)\}_{j=1}^{N}, i))) - c_i)(\Pi_i^{2})^{opnl}(\omega)$$

The utility for the ranking stage goes to zero almost surely as it only involves $\sqrt{T} + 1$ time slots and the average is taken over the total $T + 1$ time slots.

32
We now need to find the upper bound for the above utility (note that we are maximizing only at the points where the limits are defined). We use the similar approach as used in the proof of Theorem 1. Note that the above function is like a step function in terms of the variable $\Pi_i^{\text{rank}}(\omega)$. As we increase one variable say $\Pi_i^{\text{rank}}(\omega)$ either the rank of agent $i$ and hence the task stays the same for the agent or it can increase (as long as $\Pi_i^{\text{rank}} \leq e_i^{\text{max}}$). At the point at which there is a tie, if the limit of utility of agent $i$ $U_i$ exists then the limit will be bounded above and below (with probability one) by the highest and lowest tasks that can be achieved due to the tie as shown above. Hence, it is clear that if we are maximizing over the space of actions on which the limits exist, it will be best for the agent to increase the $\Pi_i^{\text{rank}}(\omega)$ as much as possible. If there is no tie in the list $\{F(j)\Pi_j^{\text{rank}}(\omega)\}_{j=1,j\neq i}^N$ with $F(i)e_i^{\text{max}}$, then $G(M^{\text{opnl}}(\{F(j)\Pi_j^{\text{rank}}(\omega)\}_{j=1,j\neq i}^N, F(i)e_i^{\text{max}}, i)$ is well defined based on assortative matching and we get the following upper bound.

$$(\alpha F(i)^2G(M^{\text{opnl}}(\{F(j)\Pi_j^{\text{rank}}(\omega)\}_{j=1,j\neq i}^N, F(i)e_i^{\text{max}}, i)) - c_i)\Pi_i^{\text{rank}}(\omega) \leq$$

$$(\alpha F(i)^2G(M^{\text{opnl}}(\{F(j)\Pi_j^{\text{rank}}(\omega)\}_{j=1,j\neq i}^N, F(i)e_i^{\text{max}}, i)) - c_i)\Pi_i^{\text{rank}}(\omega), \forall \Pi_i^{\text{rank}}(\omega) \in [0, e_i^{\text{max}}]$$

From the above inequality it is clear that $e_i^{\text{max}}$ is the dominant strategy for the agent no matter what others do.

If $(\alpha F(i)^2G(M^{\text{opnl}}(\{F(j)\Pi_j^{\text{rank}}(\omega)\}_{j=1,j\neq i}^N, i)) - c_i) > 0$, then the upper bound is given as

$$(\alpha F(i)^2G(M^{\text{opnl}}(\{F(j)\Pi_j^{\text{rank}}(\omega)\}_{j=1,j\neq i}^N, i)) - c_i)(e_i^{\text{max}})^2$$

If $(\alpha F(i)^2G(M^{\text{opnl}}(\{F(j)\Pi_j^{\text{rank}}(\omega)\}_{j=1,j\neq i}^N, i)) - c_i) \leq 0$, then the upper bound is zero.

Now we look at the utility achieved by the threshold based bang-bang learning strategy for agent $i$ and show that it can achieve the upper bound described above. Note that one crucial point about the derivation of the upper bound is that the value of $F(i)$ is assumed to be known to the agent. While the agent in our case only has an estimate that it arrives at from the first stage. The utility for our proposed threshold based strategy when all agents follow it when there are total of $T$ stages
the mean normalized output of other agents \( \{ \pi_j^T \}_{j=1}^N, \omega \mid M, P \) goes to zero as \( T \) goes to zero. Therefore, \( \lim_{T \to \infty} \frac{T}{T+1} (\alpha F(i)^2 G(M^{\text{opnl}}(\{ \hat{R}_i^T (H_i^T (\omega), \pi_i^T | M^{\text{rank}}) \}_{i=1}^N, i)) - c_i) \times \)

\((e_i^{\text{max}})^2 I(\hat{F}(i)^T (\omega) \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{ \hat{R}_i^T (H_i^T (\omega), \pi_i^T | M^{\text{rank}}) \}_{i=1}^N, i))}}) \)

Note that the above utility did not factor in the ranking stage, whose contribution anyway goes to zero as \( T \) goes to infinity. For our proposed strategy the limit of

\[ \lim_{T \to \infty} G(M^{\text{opnl}}(\{ \hat{R}_i^T (H_i^T (\omega), \pi_i^T | M^{\text{rank}}) \}_{i=1}^N, i)) = G(M^{\text{opnl}}(\{ F(j) \Pi_j^{\text{rank}} (\omega) \}_{j=1, j \neq i}^N, F(i)e_i^{\text{max}})) \]

Basically agent \( i \) will exert maximum effort and have \( F(i)e_i^{\text{max}} \) as the mean normalized output. Note that choosing the maximum effort by agent \( i \) will not affect the mean normalized output of other agents \( \{ F(j) \Pi_j^{\text{rank}} (\omega) \) as the agents have private histories. If the agents did not have private histories, then choosing highest effort may have not been the best response strategy. Next we need to compute the following limit.

\[ \lim_{T \to \infty} I(\hat{F}(i)^T (\omega) \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{ \hat{R}_i^T (H_i^T (\omega), \pi_i^T | M^{\text{rank}}) \}_{i=1}^N, i))}}) \]

If \( F(i) > \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{ F(j) \Pi_j^{\text{rank}} (\omega) \}_{j=1, j \neq i}, F(i)e_i^{\text{max}}))}} \), then

\[ I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{ F(j) \Pi_j^{\text{rank}} (\omega) \}_{j=1, j \neq i}, F(i)e_i^{\text{max}}))}} = 1. \]

We know that \( \lim_{T \to \infty} \hat{R}_j^T (H_j^T (\omega)) = F(j) \Pi_j^{\text{rank}} (\omega) \) and \( \lim_{T \to \infty} \hat{R}_i^T (H_i^T (\omega)) = F(i)e_i^{\text{max}} \) for all \( \omega \) except a set of probability zero. Therefore,

\[ \lim_{T \to \infty} I(\hat{F}(i)^T (\omega) \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{ \hat{R}_i^T (H_i^T (\omega), \pi_i^T | M^{\text{rank}}) \}_{i=1}^N, i))}}) = 1 \]

34
Similarly we can solve the case when \( F(i) < \sqrt{\frac{\alpha \sum_{j=1, j \neq i} \mu(j) \cdot \mu_i}{\alpha}} \).

Also, when \( F(i) = \sqrt{\frac{\alpha \sum_{j=1, j \neq i} \mu(j) \cdot \mu_i}{\alpha}} \) then the long-term utility is zero then the effort exerted in the operation stage does not matter.

In the derivation above we showed that the proposed threshold based bang-bang learning strategy is the best response strategy.

If all the agents use the best response strategy, then it will be a NE.

We also want to show that there is no other equilibrium in which some agent \( i \) can achieve a higher utility, which basically implies that this equilibrium is Pareto efficient when compared to any other equilibrium. To show this we will assume that \( c_i \) is lower for an agent with higher quality. We also assume that if \( F(r) e_i^{max} \geq F(j) e_j^{max} \), then \( F(r) \geq F(j) \). We first argue that if such an equilibrium exists then it has to be one in which agent \( i \) is allocated a task strictly better than \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \) (Note that this task corresponds to the one assigned by the threshold based learning strategy). If this is not the case, then clearly with a worse or the same task the agent cannot achieve a higher utility. Now suppose that there is indeed an allocation in which an agent \( i \) with quality \( F(i) \) is allocated a better task than \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \), let that task be agent \( j \)’s task from our equilibrium strategy \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \). We assume that the rank of agent \( j \) in terms of mean normalize output \( F(j) e_j^{max} \) is \( j' \). In this case it has to be that some agent \( k \) such that \( F(k) e_k^{max} \geq F(j) e_j^{max} \) (implies \( F(k) \geq F(j) \) due to the assumption) is allocated a task of quality less than the task assigned by following proposed strategy \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \). We can justify this as follows. Suppose this is not true, then note that that there will be \( N - j' \) tasks that will need to be allocated to \( N - j' + 1 \) agents (because task \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \) was assigned to agent \( i \). From pigeonhole principle we know at least one agent \( k \) will be assigned to a task lower than \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \) (matching is bijective).

If for agent \( k \) the following condition holds \( F(k)^2 G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \) > \( \frac{c_k}{\alpha} \), the agent \( k \) will have an incentive to deviate and exert effort \( e_k^{max} \), which will certainly ensure that he is allocated to at least \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \). In the other case the agent has to satisfy \( F(k)^2 G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \) \leq \( \frac{c_k}{\alpha} \) then the agent will be indifferent between task \( G(M^\text{optl}) \{ F(s) e_s^{max} \}_{s=1}^N \) and
lower task. Agent $i$ as we said is allocated to a task $G(M_{\text{opnl}}[\{F(s)_{e_{s_{\text{max}}}}\}_{s=1}^{N}, j])$.

We know that for agent $i$

$$F(i)^2G(M_{\text{opnl}}[\{F(s)_{e_{s_{\text{max}}}}\}_{s=1}^{N}, j]) \leq F(k)^2G(M_{\text{opnl}}[\{F(s)_{e_{s_{\text{max}}}}\}_{s=1}^{N}, k]) \leq \frac{c_k}{\alpha} \leq \frac{c_i}{\alpha}$$

Hence, the agent $i$ will also not exert effort even if it is allocated a better task, which implies that agent does not have a higher utility.

\[\square\]

**Theorem 4:**

- The long-term output achieved by the IILI mechanism in the TLE is the same as the total output of the FILI mechanism in TE with probability one.

- The long-term utility of the designer achieved by IILI mechanism in the TLE is the same as the long-term utility of the FILI mechanism in TE with probability one.

**Proof.** We first write the total mean output for $T + 1$ stages as follows.

$$W(\{\pi_i^T\}_{i=1}^{N} \mid M) = \frac{1}{T+1} \sum_{t=0}^{T} \sum_{i=1}^{N} W_i(H_i^t, H_0^t, \pi_i \mid M)$$

The long-term output is the almost sure limit of the above random variables. We will first compute the long-term output of the agent $i$ when all the agents follow the proposed strategy.

$$W_i(\{\pi_i^T\}_{i=1}^{N}, M) = \frac{1}{T+1} \sum_{t=\sqrt{T+1}}^{T} W_i(H_i^t, H_0^t, \pi_i^T \mid M)$$

$$= \frac{1}{T+1} \sum_{t=\sqrt{T+1}}^{T} \left( F(i)G(M_{\text{opnl}}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T \mid M_{\text{rank}})\}_{e_{i_{\text{max}}}} \times \right) I(\hat{F}(i)^{\sqrt{T}} \geq \frac{c_i}{\alpha G(M_{\text{opnl}}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T \mid M_{\text{rank}}) + Z_{iM_{\text{opnl}}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T \mid M_{\text{rank}})$

36
It is clear that the second term in the above summation will converge to zero (application of strong law of large numbers). Note that the first term of the summation does not depend on time $t$ and hence, we can simplify to obtain the following.

$$F(i)e_i^{max}G(M^{opnl}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T | M^{rank})\})I(\hat{F}(i)^{\sqrt{T}} \geq \sqrt{\frac{c_i}{\alpha G(M^{opnl}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T | M^{rank})\})})})$$

We know that $G(M^{opnl}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T | M^{rank})\}) \rightarrow G(M[\{F(s)e_s^{max}\}_{s=1}^N])$ almost surely. If $F(i)$ and $\sqrt{\frac{c_i}{\alpha G(M[\{F(s)e_s^{max}\}_{s=1}^N])})$ are distinct, then from proof of previous theorem we know that $I(\hat{F}(i)^{\sqrt{T}} \geq \sqrt{\frac{c_i}{\alpha G(M^{opnl}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T | M^{rank})\})})}) \rightarrow I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(M[\{F(s)e_s^{max}\}_{s=1}^N])})})$ almost surely. Hence, the product of the two random variables also converges almost surely. Therefore, the long-run output is given as follows.

$$W(\{\pi_i\}_{i=1}^N, M) = \sum_{i=1}^N F(i)e_i^{max}G(M[\{F(s)e_s^{max}\}_{s=1}^N])I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(M[\{F(s)e_s^{max}\}_{s=1}^N])})})$$

Next we will discuss the profit achieved by the designer. The contribution of agent $i$ to the designer’s profit is given as

$$bW_i(\{\pi_i^T\}_{i=1}^N, M) - \alpha \frac{1}{T+1} \sum_{t=\sqrt{T}}^T \frac{W_i(H_{i,1}^t, H_{i,0}, \pi_i | M)^2}{G(M^{opnl}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T | M^{rank})\}_{i=1}^N))} + \sigma_i^2 \frac{\sigma_i^2}{G(M^{opnl}(\{\hat{R}_i^{\sqrt{T}}(H_i^{\sqrt{T}}, \pi_i^T | M^{rank})\}_{i=1}^N))}$$

We can combine the result about the total output and the previous theorem to get that the above random variable takes the following value with probability 1.

$$(b - \alpha F(i)e_i^{max})F(i)e_i^{max}G(M[\{F(s)e_s^{max}\}_{s=1}^N])I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(M[\{F(s)e_s^{max}\}_{s=1}^N])})})$$
Hence, the total profit takes the same value as for the case with perfect information with probability one.

\[ \sum_{i=1}^{N} (b - \alpha F(i)e^{\max}_i) F(i)e^{\max}_i G(M[{F(s)e^{\max}_s}]_{s=1}^{N}, k) I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(M[{F(s)e^{\max}_s}]_{s=1}^{N}, k})}) \]

\[ \mathbf{Theorem 5:} \text{ The regret for every agent } i \text{ decreases as } \frac{\zeta_i}{\sqrt{T}}. \]

\[ \text{Proof.} \text{ We will compute the expected value of the utility of our approach when there are a total of } T \text{ time slots and compare with the following benchmark. In the benchmark we will assume that the agent perfectly knows its quality and tries to maximize the expected utility over the } T \text{ stages. We will start by computing the benchmark for the agent’s utility. } \]

The expected utility from the ranking stage, where the expectation is with respect to the joint distribution of the noise random variables, and its upper bound is given as

\[ \bar{U}^{\text{rank}}_i(T) = E[U^{\text{rank},T}_i(\pi_T^T|\pi_j^T, M^{\text{rank}}, P)] \]

\[ = \frac{\sqrt{T}}{T+1}E[(\alpha F(i)^2 G(M^{\text{rank}}(:, i)] - c_i)\pi_i(H^t_i)^2] \]

\[ \leq \frac{\sqrt{T}}{T+1}(\alpha F(i)^2 G(N) - c_i)(e^{\max}_i)^2I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(N)})} \]

The expected utility from the operational stage, where the expectation is with respect to the joint distribution of the noise random variables, and its upper bound is given as

\[ \bar{U}^{\text{opnl}}_i(T) = E[U^{\text{rank},T}_i(\pi_T^T|\pi_j^T, M^{\text{opnl}}, P)] \]

\[ = \frac{T - \sqrt{T}}{T+1}E[\alpha F(i)^2 G(M^{\text{opnl}}(\{\hat{R}_i^T(H^t_i, \pi_i^T)\} \{M^{\text{rank}}\})_{i=1}^{N} - c_i) \times I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{\hat{R}_i^T(H^t_i, \pi_i^T)\} M^{\text{rank}}))})] \]
Note that in the above expression the random variable inside the expectation will always increase with increase in the effort exerted ($\hat{R}_i^{\sqrt{T}}$ increases with effort exerted in the ranking stage) and as a result a better task is assigned.

We now compute the utility for the proposed threshold based bang-bang learning strategy upto time $T$.

$$\tilde{U}_{i}^{\text{rank}}(T) = \frac{\sqrt{T} + 1}{T} (F(i)^2 G[M_{\text{rank}}(:, i)] - c_i)(e^{max}_i)^2$$

Note that the value of the normalized output $\hat{R}_i^{\sqrt{T}}$ in the upper bound for operational stage given above and in our proposed strategy has to be the same.

$$\tilde{U}_{i}^{\text{opnl}}(T) = \frac{T - \sqrt{T}}{T + 1} E[\alpha F(i)^2 (G(M^{\text{opnl}}(\{\hat{R}^{\sqrt{T}}_i(H^i, \pi^T_i | M_{\text{rank}})\}) - c_i) \times $$

$$I(\hat{F}(i)^{\sqrt{T}} \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{\hat{R}^{\sqrt{T}}_i(H^i, \pi^T_i | M_{\text{rank}})\}))}}(e^{max}_i)^2$$

Now let us compute the difference between the upper bound and the proposed approach’s operational stage to get.

$$\frac{T - \sqrt{T}}{T} E[\alpha F(i)^2 (G(M^{\text{opnl}}(\{\hat{R}^{\sqrt{T}}_i(H^i, \pi^T_i | M_{\text{rank}}) - c_i)(e^{max}_i)^2$$

$$I(F(i) \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{\hat{R}^{\sqrt{T}}_i(H^i, \pi^T_i | M_{\text{rank}})\}))})$$

$$I(\hat{F}(i)^{\sqrt{T}} \geq \sqrt{\frac{c_i}{\alpha G(M^{\text{opnl}}(\{\hat{R}^{\sqrt{T}}_i(H^i, \pi^T_i | M_{\text{rank}})\}))})$$

Suppose that $\min_j |F(i) - \sqrt{\frac{c_i}{\alpha G(j)}}| > 0$ for all $i$. Therefore, there exists a lower bound $\Delta_3 > 0$ for which $\min_j |F(i) - \sqrt{\frac{c_i}{\alpha G(j)}}| \geq \Delta_3$. We can get an upper bound on the probability that $\hat{F}(i)^{\sqrt{T}}$ will be more than $\Delta_3$ distance away from its mean $F(i)$ and then use it to compute the upper bound on the regret. The upper bound on the probability will depend on the variance of $\hat{F}(i)^{\sqrt{T}}$, which we know from
In this case even though the difference of the indicator functions of the random outcomes of quality of the agent and the task. We get $\delta$ does not hold, then let us consider the case $F$. Note that we assumed here that $\beta F$. In this example we illustrate that the random matching can lead to bad outcomes.

In the above computation we assumed that $\min_j |F(i) - \sqrt{\alpha G(j)}| > 0$. If this does not hold, then let us consider the case $F(i) = \sqrt{\alpha G(M^{opt}(\sum_{j=1,...,i})}$. In this case even though the difference of the indicator functions $\left( I(F(i) \geq \sqrt{\alpha G(M^{opt}(\sum_{j=1,...,i})} - I(\hat{F}(i)) \right)^T \geq \sqrt{\alpha G(M^{opt}(\sum_{j=1,...,i})})$ in this case may not converge the regret is still zero because the utility in any period will be zero for large $T$.

7 Example

In this example we illustrate that the random matching can lead to bad outcomes. Consider a linear payment rule, which pays $\beta$ per unit output. Since the matching is random the agents will just best respond in each period. So agent $i$ has a utility in one period $\beta F(i)G(j) c_i - c_i e_i^2$. Hence, the optimal $e_i^* = \min\{e_i^{max}, \beta F(i)G(j)/2c_i\}$. The output of agent $i$ in this case is $F(i)G(j) e_i^*$. Let us assume that $\beta F(i)G(j)/2c_i < e_i^{max}$. Therefore, the output can be simplified to give $F(i)G(j)^2 \beta / 2c_i$. Now we know that matching is random. Let us compute the mean output $E[F(i)^2]G(j)^2 \beta / 2c_i$. Note that we assumed here that $\beta F(i)G(j)/2c_i < e_i^{max}$ for all the realizations of the random outcomes of quality of the agent and the task. We get $E[F(i)^2] = (q^{max})^2/3$. Substitute this $(q^{max})^2/3)G(j)^2 \beta / 2c_i < (q^{max})^2/3)s^{2/2c_i} < q^{max} e_i^{max}/3$. Thus the total output can at most be $q^{max} s^{max} e_i^{max}/3$. Using the result from Theorem 2 we have the ratio between this and the upper bound as $2/3$. In the upper bound computation $\gamma = 0$. If $\delta = 0$-homogeneous maximum effort and
$q^{max} > 1/3$, then certainly we can achieve more than the above upper bound.

References


