

Dynamic Task Assignments: An Online Two Sided Matching Approach

Sujit Gujar and Boi Faltings
Artificial Intelligence Laboratory
École polytechnique fédérale de Lausanne

Abstract

For the task assignment problem in an expert crowdsourcing platform, we propose that the dynamically arriving workers report their preferences for the tasks as ordinal preferences to the platform. We model then the task assignment problem as a dynamic two sided matching problem. In this paper we study the dynamic two sided matching when the men (the workers) side of the market is arriving dynamically and the women (the requesters) side is available since beginning. We assume strict preferences of the agents. Using a deferred acceptance algorithm as a building block, we first develop f^{APODA} , a class of strategy-proof online mechanisms. We design f^{APODA} and f^{ThODA} in this class. As no mechanism can achieve stability in our settings, we propose a weaker notion of stability, namely, progressive stability. We introduce an online mechanism f^{RODA} that achieve the progressive stability. For achieving good rank-efficiency, we design an online matching mechanism f^{BOMA} . We study all the four mechanisms empirically for stability and rank-efficiency.

1 Introduction

The term *crowdsourcing* was introduced by Howe [11]. Since then, crowdsourcing has become popular to get tasks done in the form of an open call over Internet. Currently there are thousands of websites, also called as *platforms*, available for crowdsourcing.¹ In the crowdsourcing market, there are two types of users of the platform, the one who posts the task is called *a requester* and the one who seeks to work on the task is called *a worker*. Crowdsourcing of complex macro tasks is referred to as *expert crowdsourcing* [22]. For example, oDesk, topcoder are expert crowdsourcing platforms. Consider the following scenario of an expert crowdsourcing as shown in the Figure 1.

Example 1 *On a Monday morning three requesters login to a crowdsourcing platform with their tasks. These tasks are to develop software modules and are having deadlines in two weeks. Rewards for these tasks are \$600, \$700 and \$650 respectively. m_1 , m_2 and m_3 are eligible and interested workers for these tasks. The requester of task w_1 prefers m_3 , then m_1 and then m_2 . The worker m_3 prefers to work on w_1 then m_2, m_3 . Similarly the other workers (and also the requesters) have preferences over the tasks (workers). m_1 is available from Monday morning till Tuesday evening for the task assignment, where as m_2 and m_3 are present only on Monday and Tuesday respectively. The goal is to optimally assign the tasks to the dynamic workers.*

As seen in the above example, in expert crowdsourcing market, the requesters have preferences over the workers who are assigned to their tasks and the workers have preferences over the tasks they are assigned. For the workers, one of the important advantages is to select tasks of their own choice. Hence in such task assignments, it is very important to cater to the preferences of the workers to retain them with the platform.

¹<http://crowdsourcing.org>

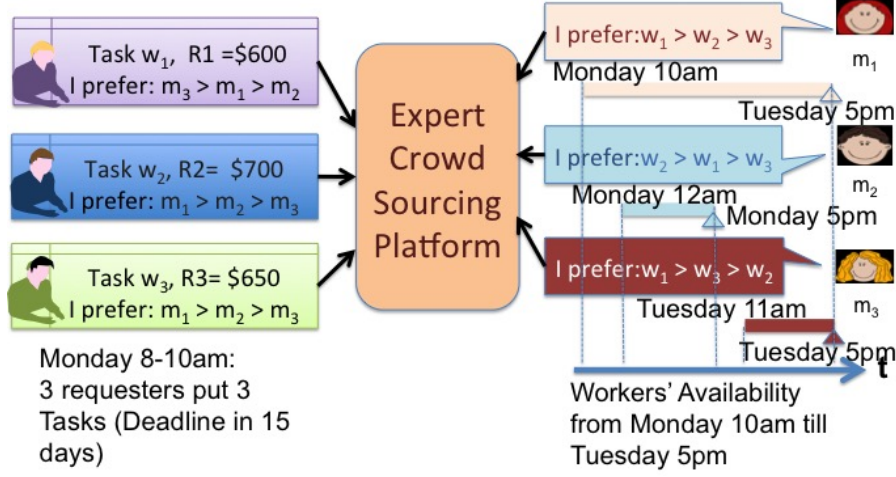


Figure 1: Crowdsourcing scenario

We model the task assignment problem as a matching in two sided market: the requesters as one side and the workers as another, both the sides having preferences for the match they obtain. We refer to the requesters as *Women* and the workers as *Men*. Two sided matching problem is extensively studied using game theory in *static settings*, that is, all men and women are simultaneously available for matching. However, this need not be the case in the real world applications. As seen in the above example, each worker arrives dynamically and needs to get the task before he leaves. Matching in such dynamic settings is called as *online matching*. The strategic men may manipulate a matching mechanism by misreporting their preferences. Gujar and Parkes [8] addressed the online matching in two sided markets in a game theoretic setting. However, in their setting men are static where as in our settings men are dynamic. The authors [8] assume external pool of men available as substitutes. In this paper we do not assume such pool of men or women and construct online matching mechanisms. In particular, the following are our contributions.

Contributions We propose to use dynamic two sided matching for the task assignment problem of an expert crowdsourcing platform.

- First we develop a class of online matching mechanisms, by partitioning dynamically arriving men, which we call as *Partition Online Deferred Acceptance*, f^{PODA} . These mechanisms are truthful.
 - We design two partition mechanisms, *Arrival Priority Online Deferred Acceptance*, f^{APODA} and *Threshold Online Deferred Acceptance* f^{ThODA} .
- It is impossible to achieve stability in online settings. Hence, we introduce a notion of *progressive stability* in online matching. We propose a f^{RODA} , an online matching mechanism that achieves the progressive stability at the cost of truthfulness. We believe that f^{RODA} satisfies weaker notion of truthfulness, namely ex-ante truthful.
- To obtain a good rank efficiency, that is average rank of a matching proposed by a mechanism, we devise an online matching mechanism f^{BOMA} .

The rest of the paper is organized as follows.

Organization First we review the related research in Section 2. In Section 3, we explain our model and notation. We design truthful mechanisms in Section 4. We propose mechanisms to improve stability and rank-efficiency in Section 5. We study our mechanisms empirically and

discuss the properties achieved by these mechanisms in Section 6. We conclude the paper in Section 7.

2 Related Work

Task Assignment in Crowdsourcing [10, 9, 4, 14, 18, 12, 22, 2, 13] addressed the task assignment problem in crowdsourcing. However, most of them are concerned only about the quality of the answers received and how to assign tasks to workers so as to meet a requester's goals. Difallah *et. al.* [6] proposed to push the tasks to the workers based on their preferences. These are only categorical preferences and not the workers' preferences for the requesters or for any specific tasks. The authors did not address the requesters' preferences. Moreover, the workers may be strategic in reporting their preferences which is not addressed in [6].

Akbarpour *et. al.* [1] designed algorithms for dynamic matching markets. The goal in [1] is to assign a maximum number of matches in large markets with dynamic population. The authors did not consider the preferences of the participants.

Two Sided Matching In their seminal work, Gale and Shapley abstracted two sided matching as a marriage problem [7]. The authors introduced a notion of *stability* and proposed an algorithm *Deferred Acceptance* (DA). Since then two sided matching problem has been extensively studied and applied in many real world settings. Roth [19] proved that there is no strategy-proof mechanism that achieves stability. Majumdar [16] proved that, under certain conditions, the DA satisfies a weaker notion of strategy-proofness, namely Bayesian Incentive Compatibility. If we allow the participants to report weak preferences, that is, indifference among alternatives, there are exponential number of stable matchings and selecting a stable matching in such settings poses algorithmic challenges. For details about static two sided matching, we refer the readers to [21, 20, 17].

In online settings, [15] designed algorithms assuming agents are truthful. Compte and Jehiel [5] considered a different dynamic matching problem to the one studied here. In their model, all men and women are static, but the men and women experience a preference shock and are interested in re-match. It imposes an individual-rationality constraint across periods so that no man or woman becomes worse off as the match changes in response to a shock. The authors demonstrated how to modify the deferred acceptance algorithm to their problem. The dynamic matching was addressed by [8] for the case of static men.

3 Preliminaries and Notation

Motivated by the task assignment problem in an expert crowdsourcing, we make certain assumptions.

Assumptions

- The men side is dynamic where as the women side is static. (In Expert crowdsourcing scenario: the tasks are complex and need much longer duration to complete them. We restrict to time window during which these tasks needs to be completed once posted. The workers log-in at different times.)
- The Men are strategic and the women are honest in reporting the preferences. (In our example, preferences of the requesters can be derived from skills required for the tasks and performance of the workers in similar tasks in the past where as the workers may be strategic in reporting their preferences.)

- The preferences of the men and women are strict.²
- The men do not lie about their arrival departure periods. Such settings are called as *exogenous*.

We use the following notation in the rest of the paper.

3.1 Notation

In the market there are n men (M) at one side and n women (W) at the other. Every agent is interested in obtaining a match at the other side. Let \succ_i be strict preference order of an agent $i \in M \cup W$ over other side of the market. A preference profile of the agents is denoted as: $\succ = (\succ_i, \succ_{-i})$, where \succ_{-i} is the preferences of all the remaining agents apart from i . m_j arrives into the market in period a_j and is available for matching till d_j . We denote the schedules of arrival and departure of men by $\rho = \{(a_j, d_j) : j \in M\}$. We denote a match by μ where $\mu(m) \in W \cup \{\emptyset\}$ and $\mu(w) \in M \cup \{\emptyset\}$. Our notation is summarized in Table 1.

n	Total number of men (women)
M	Set of Men
W	Set of Women
a_j, d_j	Arrival time and departure time for a man m_j
ρ	$= \{(a_j, d_j)_{m_j \in M}\}$ Arrival-Departure Schedule of Men
\succ_i	Preference of $i \in M \cup W$
\succ	$= (\succ_i, \succ_{-i})$ Preference profile of all agents
$W(t)$	Women that are not matched till t .
$M(t)$	$\{m_j \ni a_j \leq t \leq d_j \text{ and } m_j \text{ is not matched.}\}$
$AM(t)$	$\{m_j a_j = t\}$ Set of men arriving in time slot t
$DM(t)$	$\{m_j d_j = t\}$ Set of men departing in time slot t
f	Matching mechanism
μ	$= f(\succ, \rho)$. A matching

Table 1: Notation

An *online matching mechanism* f selects a matching $\mu = f(\succ, \rho)$ [8]. A matching mechanism f should be feasible. That is, it should match each $m_j \in M$ before d_j , that is before he leaves the system. Important desirable properties of a matching mechanism are *truthfulness*, *stability* and *rank-efficiency*.

Definition 1 (Strategy-Proof) *Online mechanism f is strategy-proof (or truthful) for men if for each man m , for all arrival-departure schedules ρ , and for all preferences \prec_{-m} , and for all $\succ'_m \neq \succ_m$,*

$$\mu'(m) \neq \mu(m),$$

$$\text{where } \mu' = f(\succ'_m, \succ_{-m}, \rho).$$

Definition 2 (Stability) *We say a pair $(m, w) \in M \times W$ blocks a matching μ if, $w \succ_m \mu(m)$ and $m \succ_w \mu(w)$. If there is no blocking pair, we say the matching is stable. And if a matching mechanism always produce stable matching, we say the mechanism is stable.*

²We further assume that the agents are able to identify their preferences. In general crowdsourcing, it may be in-feasible for agents to know own preferences over large number of tasks. However, in an expert crowdsourcing, the number of tasks in which the worker is interested, is limited.

To evaluate performance of an online mechanism, we also consider its rank-efficiency. Rank-efficiency of an online mechanism is an expected rank for each agent that it assigns to its match. We, following Budish and Cantillon [3], assume risk neutral agents with a constant difference in utility across the matches that are adjacent in their preference list. The rank of an agent i for a matching μ , written $rank_i(\mu)$, is the rank order of the agent with whom he or she is matched. A match by i with the most-preferred agent in \succ_i receives rank order 1 and with the least-preferred receives rank order n . If $\mu(i) = \emptyset$ then the rank-order is $n + 1$. Based on this, the rank of a matching μ is $rank(\mu) = \frac{1}{2n} \sum_{i \in M \cup W} rank_i(\mu)$.

To define the rank-efficiency of a mechanism we assume a distribution function Φ on (\succ, ρ) and compute the expected rank over the induced distribution on matches:

Definition 3 (Rank-efficiency) *The rank-efficiency of an online mechanism f , given distribution function Φ , is*

$$rank^f = \mathbb{E}_{(\succ, \rho) \sim \Phi} [rank(f(\succ, \rho))].$$

We first describe a celebrated algorithm, Deferred Acceptance, in the next subsection.

3.2 Static Matching Mechanisms

Lets say $\rho = \{(1, 1)_{m_i} \forall m_i \in M\}$. We refer to this as *static settings*. Gale and Shapley [7] proposed a deferred acceptance algorithm when all the agents are static.

Definition 4 (Man-proposing DA) *Each man states his most preferred woman. Each woman keeps the best match and rejects other men. All rejected men then propose to their next preferred women. The procedure continues until there are no more rejections.*

We denote matching produced by the above algorithm as $DA(M, W)$. This mechanism is strategy-proof for men and always selects a man-optimal stable matching [20]. That is no other stable matching is preferred by all the men. Similarly we can define a woman-proposing DA.³ In general, it is not possible to have a matching mechanism that is stable and strategy-proof for both sides of the market [19]. Hence, we address incentive constraints only for the men side. Since a DA has interesting game theoretic properties, we design strategy-proof mechanisms for the dynamic settings using the DA as a building block.

4 Dynamic Matching: Truthful Mechanisms

In dynamic settings, either one of the two sides is dynamic or both the sides are dynamic. In this paper, we do not address the case when both the sides are dynamic.

4.1 Dynamic Matching: The Static Men and The Dynamic Women

Gujar and Parkes [8] consider a setting when the strategic side (men) is static and the honest side (women) is dynamic. The authors propose a matching mechanism GSODAS. In GSODAS, the authors propose to use $DA(M, W(t))$ in each period t where $W(t)$ is a set of women available in t . If a man gets matched with a woman better than his previous period match, he skips the previous match. If his previously matched woman has already left the market, she gets a substitute for him. The authors assume an external pool of men available as substitutes, may be in a secondary market. The authors prove that the GSODAS is stable and strategy-proof for the men.

Though their setting resembles to our setting, in our setting the men side is dynamic where as in [8] it is static.

³Unless otherwise stated, when we say DA in this paper, we mean man-proposing DA.

4.2 Dynamic Matching: The Dynamic Men and The Static Women

As seen in Section 3, a DA is very simple, fast algorithm and always leads to stable matching in static settings. It is strategy-proof for men. Hence, we use it as a building block in designing matching mechanisms. We make some observations for dynamic settings.

(1) We do not assume possibility of substitutes. For example, consider the expert crowd-sourcing market described in the Introduction. Once a worker is assigned a task and he starts working on it, the platform cannot preempt it from him and request him to get another task in another market. Thus, if a woman w is matched with a man m and if m leaves the system in period time t , she is not available for matching for all time periods $> t$.

(2) Suppose we execute DA at t_1 and $t_2 > t_1$ with all the men available in the system at those periods. If a man m participates in both the DAs, he gets better or worse match than the first DA. Due to possible collisions across multiple DAs, it is not feasible to guarantee him the best match across multiple DAs. This can potentially create misreports by allowing him to get better match in the end. This motivates our mechanisms in the next section.

4.3 Partition Online Deferred Acceptance (PODA)

Every man is matched using DA at one period in his availability and this match is final. No man is considered for DA more than once. This induces a partition among the men based on the time of their matching. Let $\Pi = \{M_1, \dots, M_k\}$ be collection of subsets of M such that (i) $M_i \cap M_j = \emptyset \forall i \neq j$, (ii) $\cup M_i = M$, (iii) t_1, \dots, t_k such that $\forall m_j \in M_i, a_j \leq t_i \leq d_j$, and (iv) this partition is independent of the preferences of men.

With this we propose a class of online matching mechanisms which we call as *Partition online DA* (PODA).

ALGORITHM 1: Matching Mechanism f^{PODA}

Input: Preferences of Men and Women (\succ), ρ , Π , $\{t_1, \dots, t_k\}$

Output: A matching μ

```

1  $t = 1, W(1) = W$ 
2 if  $t \in \{t_1, \dots, t_k\}$  then
3    $\mu^{t_i} = DA(M_i, W(t))$ 
4  $t \leftarrow t + 1$ 
5  $W(t) \leftarrow W(t - 1) \setminus \{w : \mu^t(w) \neq \emptyset\}$ 
6 if  $W(t) == \emptyset$  then
7    $\mu = \cup_{t_i} \mu^{t_i}$ 
8   STOP.
9 GO TO STEP 2.
```

Proposition 1 f^{PODA} is strategy-proof for men.

In f^{PODA} , each man is part of a DA only once. A DA is strategy-proof for men, hence no man can benefit by misreporting his preference when he is matched using a DA. As the partition is independent of the preferences of the men, no man's preference can influence his competitors in DA. That implies that no man can benefit by misreporting. Hence, f^{PODA} is strategy-proof for men.

□

Examples of f^{PODA}

f^{PODA} assumes Π is given. We show with two examples how to create partition without knowing any details about the men arriving in the future and their schedule.

4.3.1 Arrival Priority Online Deferred Acceptance (f^{APODA})

In this mechanism we consider the partition of men induced by their arrival periods. That is, in each period t if $AM(t) \neq \emptyset$ we execute $DA(AM(t), W(t))$. Its very simple and easy to implement. However, it may lead to many blocking pairs. To optimize for stability, we propose the following mechanism.

4.3.2 Threshold Online Deferred Acceptance (f^{ThODA})

In this mechanism, we induce a partition of the men greedily and based on a given parameter threshold (Th). The basic idea is to accumulate more men for matching every time we execute the DA. The parameter Th can be optimized for a given stochastic process of arrival-departure of the men. The proposed matching mechanism is follows:

ALGORITHM 2: Matching Mechanism f^{ThODA}

Input: Preferences of Men and Women (\succ), ρ , Th

Output: A matching μ

```

1  $t = 1$ ,  $M(1) = AM(1)$ ,  $W(1) = W$ 
2 if  $DM(t) == \emptyset$  then
3    $\quad$  DO NOTHING
4 if  $DM(t) \neq \emptyset$  and  $|M(t)| > Th$  then
5    $\quad \mu^t = DA(M(t), W(t))$ 
6 if  $DW(t) \neq \emptyset$  and  $|M(t)| \leq Th$  then
7    $\quad \mu^t = DA(DM(t), W(t))$ 
8  $t \leftarrow t + 1$ 
9  $W(t) \leftarrow W(t-1) \setminus \{w : \mu^t(w) \neq \emptyset\}$ ,
    $M(t) \leftarrow \{M(t-1) \cup AM(t)\} \setminus \{m : \mu^t(m) \neq \emptyset\} \cup DM(t)$ 
10 if  $W(t) == \emptyset$  then
11    $\quad \mu = \cup_t \mu^t$ 
12    $\quad$  STOP.
13 GO TO STEP 2.
```

Both the above mechanisms are truthful. We now discuss the stability of these mechanisms in the following subsection.

4.4 Stability in Partition DA

In general, no mechanism can predict preferences of the agents yet to arrive. So, it is not fair to expect stability in dynamic settings. Note that GSODAS can achieve stability with external pool of men in secondary markets. We do not assume possibility of substitutes. It follows from Proposition 3.1 in [8] that stability is impossible in our settings. Hence, in this paper, we propose a weaker notion of stability, namely *progressive stability*. The idea is, at each instance of time, there is no blocking pair in the system. We do not allow women, that are matched with men who are no longer available for matching, to form a blocking pair. More formally, progressive stability is defined as follows.

Definition 5 (Progressive Stability) A pair (m, w) is said to be blocking pair at time t if (i) m, w both are present in the system at t , and not matched with each other, (ii) prefer to match with each other than their current match, (iii) their current matches are also present in the system. In each time period, if no such pair exists, we say a matching is progressively stable.

PODA mechanisms cannot achieve progressive stability. The partition is independent of the preferences. Consider the following instance of preferences. Let $t_1 < t_2$ and w matched with $m_1 \in M_1$ with $d_1 > t_2$. For some $m_2 \in M_2$, if $m_2 \succ_w m_1$ and m_2 prefers w most, (m_2, w) is blocking pair at t_2 .

In the next section, we look for non-strategy proof mechanisms to improve stability and rank-efficiency.

5 Dynamic Mechanisms: Progressive Stability and Rank-Efficiency

5.1 Repeated Online Deferred Acceptance (RODA)

We run the DA in every period and only the matches involving the departing men are final.

ALGORITHM 3: Matching Mechanism f^{RODA}

Input: Preferences of Men and Women $(\succ), \rho$

Output: A matching μ

```

1  $t = 1, M(1) = AM(1), W(1) = W$ 
2  $\mu^t = DA(M(t), W(t))$ 
3 for  $m \in DM(t)$  do
4    $\mu(m) = \mu^t(m)$  and  $\mu(\mu^t(m)) = m$ 
5  $t \leftarrow t + 1$ 
6  $M(t) \leftarrow \{M(t-1) \cup AM(t)\} \setminus DM(t), W(t) \leftarrow \{W(t-1) \setminus \{w : \mu^t(w) \in DM(t)\}\}$ 
7 if  $W(t) == \emptyset$  then
8   STOP.
9 GO TO STEP 2.
```

In f^{RODA} , men get match only at their departure. Hence, men may have strong incentive for early departure. However, in this paper, we focus only on exogenous settings. We illustrate f^{RODA} with the following example.

Example 2 Consider the example shown in Figure 1. Let their preferences be as in Table 2. We refer to Monday as $t = 1$ and Tuesday as $t = 2$. Workers m_1, m_2 arrive at $t = 1$ and m_2 leaves in the same slot. m_3 arrives at $t = 2$. In first round, (m_1, w_1) and (m_2, w_2) are matched using the DA. As m_2 has to leave by end of slot 1, that match is final. m_1 has deadline $t = 2$ and hence does not start working on T_1 . In second period, using the DA (m_1, w_3) and (m_3, w_1) are matched and as all workers have appeared in the system, this match is final.

Workers (Men)		Requesters (Women)	
m_1	$w_1 \succ w_2 \succ w_3$	w_1	$m_3 \succ m_1 \succ m_2$
m_2	$w_2 \succ w_1 \succ w_3$	w_2	$m_1 \succ m_2 \succ m_3$
m_3	$w_1 \succ w_3 \succ w_2$	w_3	$m_1 \succ m_2 \succ m_3$

Table 2: f^{RODA} Example Preferences

Theorem 5.1 f^{RODA} is progressively stable.

Proof: We prove by induction. At $t = 1$, we are using the static version of a DA and hence the matching at $t = 1$ are stable. Say f^{RODA} is progressively stable till $t = \tau$. We show that f^{RODA} cannot introduce any blocking pair at $t = \tau + 1$. Let's assume (m, w) is blocking at $\tau + 1$. Say at $\tau + 1$, m is matched with w' and w is matched with m' . For $m : w \succ w'$, and for $w : m \succ m'$. In the time slot $\tau + 1$, m must have proposed to w before w' . If w is matched with m' from previous round and is still present in the system, w would have rejected him and be matched with m . Thus, w must be matched with some worker who already left the system. Hence, w cannot be part of blocking pair in progressive stability. Thus, f^{RODA} is progressively stable. \square

f^{RODA} achieves progressive stability but is manipulable:

Claim 1 f^{RODA} is not strategy-proof.

Proof: In Example (2), m_1 can report preference to be $w_2 \succ w_1 \succ w_3$. He gets matched with w_2 in period 1 and this does not change in period 2. By misreporting, he obtains a preferable match than w_3 . \square

Note that, (i) such manipulation requires information about the preference of a man yet to arrive. (ii) By simulations with uniform preferences, we observed, $\text{Prob}(\text{rank}_i(f^{RODA}(\succ, \rho)) = k)$ decreases with k . Hence we believe that f^{RODA} is ex-ante strategy-proof for uniform preferences. (iii) If beliefs of men about the preferences of men yet to arrive are uniform even after observing their own preference, it need not give any incentive to men to misreport in f^{RODA} .

5.2 Rank-Efficiency

In the static settings, the DA selects a man-optimal stable matching and hence, it is Pareto-efficient. However, as we consider ordinal preferences, to measure efficiency, we use a rank-efficiency of a matching mechanism. To achieve stability, the DA mechanism has to compromise on rank-efficiency.

For example, say there are three men m_1, m_2, m_3 and three women w_1, w_2, w_3 . Let $\succ_{m_1} = w_1 \succ w_2 \succ w_3$ and $\succ_{m_2} = \succ_{m_3} = w_1 \succ w_3 \succ w_2$. All three women have preference $m_1 \succ m_2 \succ m_3$. Assume $a_m = d_m = 1$ for all three men. The DA will produce matching $\mu^D : (m_1, w_1), (m_2, w_3), (m_3, w_2)$ with $\text{rank}(\mu^D) = 2$ and a matching $\mu : (m_1, w_2), (m_2, w_1), (m_3, w_3)$ has $\text{rank}(\mu) = \frac{11}{6}$.

With this example, to improve rank-efficiency, we propose f^{BOMA} , an online matching mechanism that uses maximum weight bipartite matching for matching the men.

5.2.1 Bipartite Online Matching Algorithm

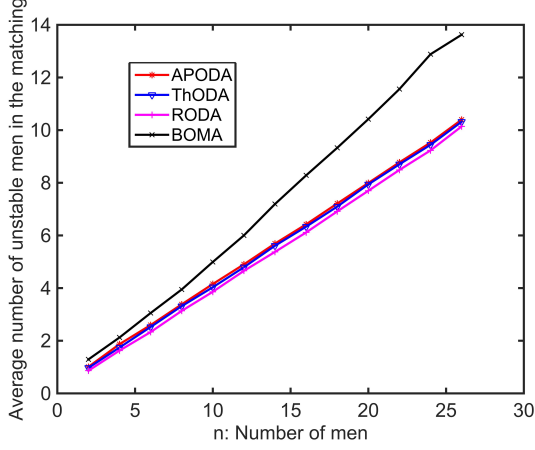
f^{BOMA} is same as f^{ThODA} except that in lines 5 and 7 of Algorithm 2, we use $\text{Max-wt-Bipartite}(M(t), W(t))$ and $\text{Max-wt-Bipartite}(DM(t), W(t))$ respectively instead of the DA. $\text{Max-wt-Bipartite}(A, B)$ gives maximum weight bipartite matching between men A and women B . An edge between a man m and woman w has weight $2n + 2 - \text{rank}_m(w) - \text{rank}_w(m)$.

We now perform an empirical study of the proposed mechanisms in the next section.

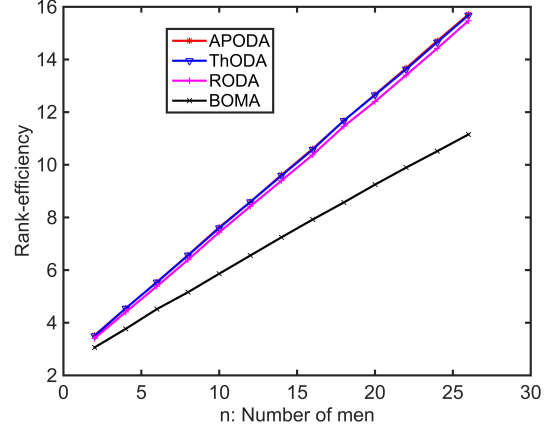
6 Evaluation of the Mechanisms

6.1 Empirical Evaluation

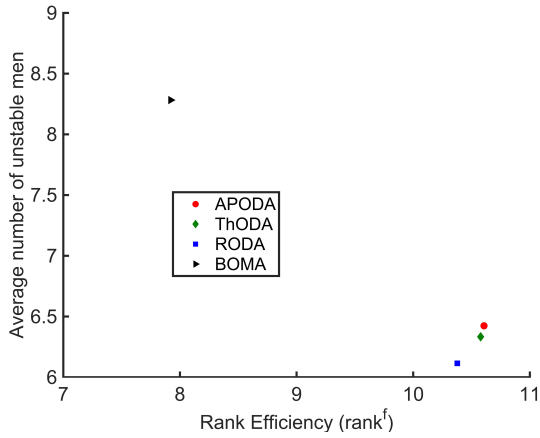
In this section, we describe the simulations that we carried to measure stability and rank-efficiency of the proposed mechanisms. We assume the men arrive into the system according a



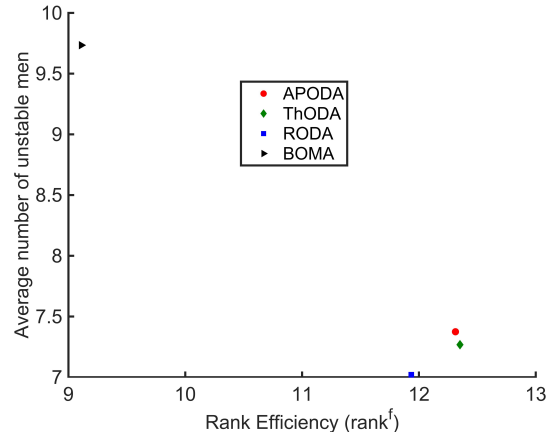
(a) Stability of the four mechanisms for various n with $\lambda = 3, \mu = 0.5$



(b) Rank-efficiency of the four mechanisms for various sn with $\lambda = 3, \mu = 0.5$



(c) Scatter Plot for rank-efficiency and stability of the four mechanisms for $n = 20, \lambda = 3, \mu = .05$



(d) Scatter Plot for rank-efficiency and stability of the four mechanisms for $n = 24, \lambda = 5, \mu = .05$

Figure 2: Comparison of $f^{APODA}, f^{ThODA}, f^{RODA}, f^{BOMA}$

Poisson process with parameter λ and wait in the system according to a exponential distribution with mean $\mu = 0.5$. The preferences of every man and woman are drawn uniformly at random from all possible preference profiles. With these setup, we generated 5,000 instances of matching problem for each of various n, λ combinations and measured stability and rank-efficiency across these instances.

To study stability, we measure the average number of unstable men produced by the mechanism. We say a man is *unstable* if he is involved in at least one blocking pair.

Figure 2 (a) and (b) shows how the stability and rank-efficiency changes for all the mechanism as we increase n by fixing $\lambda = 3$ Figure 2 (c) is a scatter plot for stability vs rank-efficiency of all the mechanisms for $n = 20, \lambda = 3$ and (d) is a scatter plot for $n = 24, \lambda = 6$.

We discuss compare all the proposed mechanisms in the next subsection.

6.2 Discussion

We designed four mechanisms for two sided matching problem when the men side is dynamic. Based on a design goal, we can choose which mechanism to be used. If strategy-proofness is important, we propose to use any mechanism from a class f^{PODA} . These are partition mechanism in which a feasible partition of the men is given. If men are impatient or if it is difficult to know how to partition, one can use $f^{APODA} \in f^{PODA}$. If underlying stochastic model is available, we

propose to use $f^{ThODA} \in f^{PODA}$. It improves the stability by 2%-5% over f^{APODA} .

If stability is of utmost important, we recommend to use f^{RODA} which is progressively stable and performs better than other mechanisms on conventional stability. If the design goal is to improve rank-efficiency, we recommend to use f^{BOMA} that improves rank-efficiency by 25% but the average number of unstable men increases by 35%. This is summarized in Table 3. The numbers in a row indicate ranking among the four mechanisms on respective performance measure.

	f^{APODA}	f^{ThODA}	f^{RODA}	f^{BOMA}
Strategy-proof	Y	Y	N	N
Stability	3	2	1	4
Progressive Stability	N	N	Y	N
Rank-efficiency	3	4	2	1

Table 3: Comparison of f^{APODA} , f^{ThODA} , f^{RODA} , f^{BOMA}

7 Conclusion

Motivated by an expert crowdsourcing market, in this paper, we addressed two sided dynamic matching problem when one side, the men side, is dynamically arriving to the market where as the other side, the women side, is available for matching from the start. We focused on exogenous men settings. We first proposed strategy-proof mechanisms, f^{PODA} , f^{APODA} , f^{ThODA} . As it is impossible to achieve stability in online settings, we introduced a weaker notion of stability, namely progressive stability. We proposed a mechanism f^{RODA} that achieves the progressive stability if all the agents are truthful. However, f^{RODA} is not strategy-proof. We also proposed a mechanism f^{BOMA} to improve the rank-efficiency, but it has poor stability and is not strategy-proof. In the previous section we compared all the mechanisms empirically. Based on a design goal, one can choose an appropriate matching mechanism.

References

- [1] Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Dynamic matching market design. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 355–355. ACM, 2014.
- [2] Satyanath Bhat, Swaprava Nath, Onno Zoeter, Sujit Gujar, Yadati Narahari, and Chris Dance. A mechanism to optimally balance cost and quality of labeling tasks outsourced to strategic agents. In *Thirteenth International Conference on Autonomous Agents and Multiagent Systems*, pages 917–924, 2014.
- [3] E. Budish and E. Cantillon. Strategic behavior in multi-unit assignment problems: Lessons for market design. Technical report, Harvard Business School, 2009.
- [4] Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 64–72, 2013.
- [5] O. Compte and P. Jehiel. Voluntary participation and re-assignment in two-sided matching. Working papers, Paris School of Economics, October 2008.
- [6] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Pick-a-crowd: tell me what you like, and i’ll tell you what to do. In *Proceedings of the 22nd international*

- conference on World Wide Web*, pages 367–374. International World Wide Web Conferences Steering Committee, 2013.
- [7] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, pages 9–15, 1962.
 - [8] Sujit Gujar and David C Parkes. Dynamic matching with a fall-back option. In *ECAI*, pages 263–268, 2010.
 - [9] Chien-Ju Ho, Shahin Jabbari, and Jennifer W Vaughan. Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 534–542, 2013.
 - [10] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.
 - [11] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.
 - [12] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation, HCOMP '10*, pages 64–67, New York, NY, USA, 2010. ACM.
 - [13] Shweta Jain, Sujit Gujar, Onno Zoeter, and Y. Narahari. A quality assuring multi-armed bandit crowdsourcing mechanism with incentive compatible learning. In *Thirteenth International Conference on Autonomous Agents and Multiagent Systems*, pages 1609–1610, 2014.
 - [14] David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 284–291. IEEE, 2011.
 - [15] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proc. STOC'90*, pages 352–358, 1990.
 - [16] Dipjyoti Majumdar. Ordinally bayesian incentive compatible stable matchings. Working Papers 05001, Concordia University, Department of Economics, August 2003.
 - [17] David Manlove. *Algorithmics of matching under preferences*. World Scientific Publishing, 2013.
 - [18] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *J. Mach. Learn. Res.*, 11:1297–1322, August 2010.
 - [19] Alvin E Roth. The economics of matching: Stability and incentives. *Mathematics of operations research*, 7(4):617–628, 1982.
 - [20] Alvin E Roth and Marilda A Oliveira Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*. Number 18. Cambridge University Press, 1992.
 - [21] Tayfun Sönmez and M Utku Ünver. Matching, allocation, and exchange of discrete resources. *Handbook of Social Economics*, 1:781–852, 2011.
 - [22] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *European Conference on Artificial Intelligence*, pages 768–773, 2012.