Necessarily Optimal One-Sided Matchings

Hadi Hosseini,[1] **Vijay Menon,**[2] **Nisarg Shah,**[3] **Sujoy Sikdar** [4]
[1] College of Information Sciences and Technology, Penn State University
[2] David R. Cheriton School of Computer Science, University of Waterloo
[3] Department of Computer Science, University of Toronto
[4] Department of Computer Science, Binghamton University
hadi@psu.edu, vijay.menon@uwaterloo.ca, nisarg@cs.toronto.edu, ssikdar@binghamton.edu

Abstract

We study the classical problem of matching n agents to n objects, where the agents have ranked preferences over the objects. We focus on two popular desiderata from the matching literature: Pareto optimality and rank-maximality. Instead of asking the agents to report their complete preferences, our goal is to learn a desirable matching from partial preferences, specifically a matching that is necessarily Pareto optimal (NPO) or necessarily rank-maximal (NRM) under any completion of the partial preferences. We focus on the top-k model in which agents reveal a prefix of their preference rankings. We design efficient algorithms to check if a given matching is NPO or NRM, and to check whether such a matching exists given top-k partial preferences. We also study online algorithms for eliciting partial preferences adaptively, and prove bounds on their competitive ratio.

1 Introduction

Resource allocation is a fundamental problem in artificial intelligence and multi-agent systems. One particular type of resource allocation deals with assigning a number of indivisible objects to agents according to their preferences. These problems have given rise to a wide array of research in economics (Moulin 2004) and theoretical computer science (Manlove 2013).

The focus of our work is the special case of allocating n objects to n agents (so each agent is matched to a single object), which models many real-world applications (see, e.g., (Hylland and Zeckhauser 1979; Abdulkadiroğlu and Sönmez 1999)). For instance, imagine allocating office spaces to faculty members in a building. Instead of asking each faculty member to report a full preference ranking over the available offices, the department head may ask them to reveal their top choices, and then, if need be, she may ask individual faculty members to reveal their next best choices, and so on. The goal is to come up with a matching that *necessarily* satisfies some form of "economic efficiency" while asking as few queries as possible. That is, it must satisfy the economic efficiency property regardless of the parts of the preferences that were not elicited, because otherwise some faculty may find the final matching undesirable.

What form of economic efficiency might one want for a matching? Luckily, decades of research in matching theory offers two prominent desiderata: *Pareto optimality* (Shapley and Scarf 1974; Cirillo 2012) and *rank-maximality* (Irving 2003; Irving et al. 2006). Informally, Pareto optimality requires that no other matching be able to make some agents better off without making any agent worse off. Despite its wide use, Pareto optimality may be a weak guarantee in many practical settings. Hence, a stronger guarantee called rank-maximality is often used in practical applications such as assigning papers to referees (Garg et al. 2010), assigning rented resources to customers (Abraham et al. 2006), and assigning students to schools (Abraham 2009). Informally, rank-maximality requires matching as many agents to their top choice as possible, subject to that matching as many agents to their second choice as possible, and so on.

The study of these axioms, along with axioms of fairness and incentive-compatibility, has led to the design of elegant rules such as random priority and probabilistic serial (Bogomolnaia and Moulin 2001; Che and Kojima 2010), which have been made easily accessible by not-for-profit endeavors such as MatchU.ai (www.matchu.ai). However, in many real-life situations, one rarely has access to complete preferences which can simply be fed to these rules. More often than not, the problem at hand is to come up with a desirable matching of agents to objects with only partial information about agents' preferences. While the role of partial preferences has been well-explored in the sister problem known as *two-sided matching* (Rastegari et al. 2013; Liu et al. 2014), in which agents are matched to other agents (for example matching roommates, kidney donors to patients, or men to women), it has been significantly understudied for one-sided matching. This is the primary focus of our work. In particular, we consider the following three research questions.

> *Given partial preferences, can we efficiently check if a given matching is NPO or NRM? Can we efficiently check if an NPO or NRM matching exists (and if so, find one)? How much online elicitation of preferences is needed to find an NPO or NRM matching?*

Our Results

We study the *next-best* query model which allows, in each query, asking one agent to reveal her next best choice. Partial preferences elicited under this model are referred to as *top-k* preferences as each agent reveals a prefix of her preference ranking consisting of the top k items (where k can be different for each agent). As mentioned above, our goal is to design online algorithms that ask few queries and learn a matching that is necessarily Pareto optimal or necessarily rank-maximal w.r.t. the elicited partial preferences. We study their performance in terms of their competitive ratio (Borodin and El-Yaniv 2005).

Note that these algorithms need to check if there already exists a matching that is NPO or NRM w.r.t. the elicited preferences in order to decide if further elicitation is required. Along the way, it is also useful to know if a *given* matching is NPO or NRM w.r.t. the given partial preferences. Thus, we also study these two questions for NPO and NRM.

For necessary Pareto optimality, we show that checking whether a given matching is NPO reduces to computing the maximum weight matching in a bipartite graph, which can be solved efficiently. We also show that there exists an NPO matching if and only if there exists a matching that matches at least $n-1$ agents to objects they have revealed, where n is the number of agents; this can be checked in polynomial-time, and if this is the case, our algorithm efficiently finds an NPO matching. Our main result is an online algorithm for finding an NPO matching that has a competitive ratio of $O(\sqrt{n})$. By proving a matching lower bound, we show that our algorithm is asymptotically optimal.

For necessary rank-maximality, we design efficient algorithms for checking whether a given matching is NRM. For checking whether an NRM matching exists; unlike in the case of NPO, these questions do not seem to reduce to checking a simple analytical condition. In the supplementary material, we show that a simple characterization can be obtained in the special case where each agent has revealed exactly the same number of objects. For online elicitation, we show a constant lower bound on the competitive ratio, and conjecture that the optimal competitive ratio is in fact constant, unlike in the case of NPO.

Related Work

The problem of matching n agents to n objects models a number of scenarios and is well-studied, sometimes as the house allocation or one-to-one object allocation problem (e.g., (Hylland and Zeckhauser 1979; Abdulkadiroğlu and Sönmez 1998; Abraham et al. 2004; Irving 2003)). There are two approaches to learning a desirable matching given only partial information about agents' preferences. One approach is to consider the worst case scenario over missing preferences; imposing a property X in the worst case leads to the "necessarily X" concept. This concept has been considered in other contexts such as in voting (Xia and Conitzer 2011) and in one-sided matching problems where an agent may be assigned to more than one good (Bouveret, Endriss, and Lang 2010; Aziz et al. 2015, 2019b). However, in the latter case, even having access to agents' full preference rankings over individual goods is not sufficient for checking Pareto optimality, which leads Aziz et al. (2019b) to consider NPO given full rankings. In contrast, in our one-to-one matching setting, full preference rankings over individual goods are sufficient to check whether a given matching is PO, which leads us to consider partial (top-k) rankings.

Another approach is to assume that the underlying preferences are drawn from a known distribution (Rastegari et al. 2013), and consider the probability of satisfying X (Aziz et al. 2019a). We note that asking whether a matching satisfies X with probability 1 often coincides with the "necessarily X" concept.

We focus on the top-k model of partial preferences, which is commonly used in the literature (Aziz, Walsh, and Xia 2015; Drummond and Boutilier 2013). However, there are also other interesting models; one such model is where the information available regarding each agent's preferences is in the form of a set of pairwise comparisons among objects (Aziz et al. 2019a). In general, the goal of this line of work is to make matching algorithms more practically relevant, to which our work contributes.

Other than economic efficiency, researchers have also focused on notions of fairness and incentive-compatibility in matching agents to objects, which has led to the design of rules such as random priority and probabilistic serial (Bogomolnaia and Moulin 2001; Abdulkadiroğlu and Sönmez 1998; Hosseini, Larson, and Cohen 2018). A related line of research considers two-sided matching markets, where agents are matched to other agents. Working with partial preferences is well-explored in such markets (Rastegari et al. 2013; Liu et al. 2014).

2 Preliminaries

Let $N = \{a_1, \cdots, a_n\}$ denote the set of agents and $O = \{o_1, \cdots, o_n\}$ be the set of objects. We use $[k]$ to denote the set $\{1, \ldots, k\}$. Thus, for $i \in [n]$, a_i and o_i represent agent and object i respectively. A *matching* of agents to objects is a bijection $M : N \to O$. We sometimes refer to a matching as a collection of pairs $\{(a_i, M(a_i) \mid i \in [n])\}$ and use $\mathcal{M}$ to denote the set of all possible matchings.

Each agent $a_i \in N$ has an underlying linear order R_i over O, where a linear order over O is a transitive, antisymmetric, and total relation on O. We use $\mathcal{L}(O)$ to denote the set of all linear orders over O, and refer to $R = (R_1, \cdots, R_n) \in \mathcal{L}(O)^n$ as a *(complete) preference profile*. Additionally, for $i, j, k \in [n]$, if a_i prefers o_j to o_k according to R_i, then we denote this by $o_j \succ_{R_i} o_k$ or $o_k \prec_{R_i} o_j$.

We formally describe two notions of economic efficiency, namely Pareto optimality and rank-maximality.

Definition 1 (Pareto optimality)**.** Given a preference profile $R = (R_1, \ldots, R_n)$, a matching M is said to be Pareto optimal w.r.t. R if no other matching can make an agent happier without making some other agent less happy, i.e., if $\forall M' \in \mathcal{M} : \ (\exists a_i \in N, M'(a_i) \succ_{R_i} M(a_i)) \Rightarrow$ $\left(\exists a_j \in N, M'(a_j) \prec_{R_j} M(a_j)\right)$.

Definition 2 (rank-maximality)**.** Given a preference profile R and a matching M, let the *signature* of M w.r.t. R, denoted $\text{sig}_R(M)$, be the tuple $(x_1, \ldots, x_n)$, where x_ℓ is the number of $(a_i, o_j) \in M$ such that o_j is the ℓ^{th} most preferred choice of a_i. We say that $(x_1, \ldots, x_n) \succ_{\text{sig}} (x'_1, \ldots, x'_n)$ if $\exists k \in [n]$ such that $x_k > x'_k$ and $x_i = x'_i$ for all $i < k$; this is the lexicographic comparison of signatures. A matching

M is rank-maximal w.r.t. R if $\text{sig}_R(M)$ is maximum under $\succ_{\text{sig}}$. In words, a matching is rank-maximal if it matches max. number of agents to their top choice, subject to that the max. number of agents to their second choice, and so on.

Partial Preferences

In this paper, we study a partial preference model called *top-k preferences*, in which each agent i may reveal a prefix of her preference ranking, that is, a ranking of her k_i most favorite objects for some $k_i \in [n]$. Specifically, agent i may reveal P_i which is a linear order over some $L \subseteq O$ such that any object in L are preferred to any object in $L \setminus O$ by the agent; we say that agent i has revealed object o in P_i if $o \in L$. Also, we sometimes use $|P_i| = |L| = k_i$ to denote the size of P_i, i.e., the number of objects revealed in P_i. We refer to $P = (P_1, \ldots, P_n)$ as a top-k preference profile and say that a linear order R_i is *consistent* with P_i if, for all $j, k \in [n]$, $o_j \succ_{P_i} o_k \Rightarrow o_j \succ_{R_i} o_k$. We use $\mathcal{C}(P_i)$ to denote the set of linear orders over O that are consistent with P_i, and $\mathcal{C}(P)$ to denote $\mathcal{C}(P_1) \times \ldots \times \mathcal{C}(P_n)$.

Given a top-k preference profile $P = (P_1, \cdots, P_n)$, we use $\text{rev}(P)$ to denote the set of all pairs (a_i, o_j) such that a_i has revealed object o_j in P_i, and $\text{unrev}(P)$ to denote $(N \times O) \setminus \text{rev}(P)$. Throughout, whenever we refer to the size of a matching M w.r.t. to P, we mean $|M \cap \text{rev}(P)|$.

We are interested in the notion of "necessarily efficient" matchings, which are guaranteed to satisfy an efficiency notion such as Pareto optimality or rank-maximality given only a top-k preference profile (Gaspers et al. 2014; Xia and Conitzer 2011; Aziz, Walsh, and Xia 2015).

Definition 3 (necessary Pareto optimality (NPO) and necessary rank-maximality (NRM))**.** Given a top-k preference profile P, a matching M is said to be necessarily Pareto optimal (resp. necessarily rank-maximal) w.r.t. P if it is Pareto optimal (resp. rank-maximal) w.r.t. any profile of linear orders that is consistent with P, i.e., for any $R \in \mathcal{C}(P)$.

Query Model and Elicitation

We consider algorithms that can elicit preferences by asking agents to reveal their "next-best" objects—meaning if the agent has already revealed their $k-1$ most preferred objects, then it asks the agent to reveal their k-th most preferred object. We formally define this query model below.

Definition 4 (next-best query model)**.** For $a_i \in N$ and $k \in [n]$, query $\mathcal{Q}(a_i, k)$, which asks agent a_i to reveal the object in the k-th position in her preference list, is a valid query in the next-best query model if query $\mathcal{Q}(a_i, j)$ has already been made for all $j \in [k-1]$.

We are interested in *online* elicitation algorithms that find NPO or NRM matchings in the next-best query model and measure their performance against an all-powerful optimal algorithm. An all-powerful optimal algorithm is a hypothetical offline algorithm that can identify the minimum number of queries of the form $\mathcal{Q}(a_i, j)$ that are sufficient to guarantee an NPO or NRM matching. That is, for $i \in [n]$ and $k_i \in [n]$, the optimal algorithm can minimize $\sum_{i \in [n]} k_i$, and can claim that asking every agent i for their top-k_i order, P_i, is sufficient to compute a matching that is NPO or NRM w.r.t. all consistent completions of $P = (P_1, \ldots, P_n)$. We say that an online elicitation algorithm is *α-competitive* (or, equivalently, it achieves a competitive ratio of α) if the number of queries it requires to ask in the worst-case across all possible instances is at most $\alpha \cdot \text{OPT}$, where OPT is the number of queries asked by the optimal algorithm.

3 Necessarily Pareto Optimal Matchings

We begin, in Section 3, by presenting an algorithm that determines if a given matching is NPO w.r.t. to a profile of top-k preferences. Following this, we provide a useful characterization for when NPO matchings exist. This in turn is used in Section 3 in designing an elicitation algorithm that is $2(\sqrt{n}+1)$-competitive. We also show that no elicitation algorithm can achieve a competitive ratio of $o(\sqrt{n})$, establishing our algorithm as asymptotically optimal.

Determining Whether a Matching is NPO

We present a polynomial time algorithm to determine whether a given matching M is NPO w.r.t. a profile of top-k preferences P. The algorithm proceeds in two steps: In Step 1, we build a directed graph $\mathbb{G} = (N, E)$, where E consists of directed edges (a_i, a_j) such that $M(a_j) \succ_{R_i} M(a_i)$ under some $R_i \in \mathcal{C}(P_i)$. That is, there exists an edge when agent i prefers the object allocated to agent j over her own under some completion of her partial preferences. In Step 2, we efficiently check if this directed graph admits a cycle. If it does, matching M is not NPO; otherwise, it is NPO.

Theorem 1. *Given a top-k preference profile P and a matching M, there is a polynomial time algorithm to check if M is NPO w.r.t. P.*

Proof. We claim that the algorithm described above—formally presented as Algorithm 1 in the full version (Hosseini et al. 2020)—is such an algorithm. First, let us prove its correctness. The key observation, which the reader can easily verify, is that the graph $\mathbb{G}$ constructed in Step 1 contains an edge (a_i, a_j) if and only if there exists a completion $R_i \in \mathcal{C}(P_i)$ under which $M(a_j) \succ_{R_i} M(a_i)$. This includes edges (a_i, a_j) for which $M(a_j) \succ_{P_i} M(a_i)$ is already true under the top-k preference order P_i.

Suppose the algorithm returns NO. Then, $\mathbb{G}$ has a cycle. Let C be one such cycle. Construct a matching, M', from M as follows: if agent i has an outgoing edge $(a_i, a_j) \in C$ in the cycle, set $M'(a_i) = M(a_j)$, and if agent i is not part of cycle C, then set $M'(a_i) = M(a_i)$. It is easy to verify that M' is a matching.

Let $R = (R_1, \ldots, R_n) \in \mathcal{C}(P)$ be a completion such that if agent i has an outgoing edge $(a_i, a_j) \in C$ in the cycle, then $R_i \in \mathcal{C}(P_i)$ satisfies $M'(a_i) = M(a_j) \succ_{R_i} M(a_i)$, and otherwise $R_i \in \mathcal{C}(P_i)$ is chosen arbitrarily. Then, it is easy to see that for each agent i who is a part of cycle C, we have $M'(a_i) \succ_{R_i} M(a_i)$, and for every remaining agent i, we have $M'(a_i) = M(a_i)$. Thus, M' Pareto-dominates M under the completion $R \in \mathcal{C}(P)$, which implies that M is not NPO w.r.t. P, as desired.

Conversely, suppose M is not NPO w.r.t. P. Hence, there exists a matching M' and a completion $R \in \mathcal{C}(P)$ such that M' Pareto-dominates M under R. We claim that $\mathbb{G}$ must have a cycle, and the algorithm therefore must return NO. To see this, consider the graph $\mathbb{G}' = (N, E')$ where, for $a_i \neq a_j$, $(a_i, a_j) \in E'$, if M' assigns to agent i the object that was assigned to agent j under M, i.e., if $M'(a_i) = M(a_j) \neq M(a_i)$. Note that in $\mathbb{G}'$, each agent i either has exactly one incoming and one outgoing edge, or no incident edges. Hence, $\mathbb{G}'$ is a union of disjoint cycles. In particular, $\mathbb{G}'$ has at least one cycle.

However, since M' Pareto-dominates M under R, for every edge $(a_i, a_j) \in E'$, we have that $M'(a_i) = M(a_j) \succ_{R_i} M(a_i)$. Hence, by the observation above, edge (a_i, a_j) must also exist in $\mathbb{G}$. Thus, $\mathbb{G}'$ is a subgraph of $\mathbb{G}$. Since $\mathbb{G}'$ has at least one cycle, so does $\mathbb{G}$, as desired.

Finally, Algorithm 5 runs in polynomial time because constructing $\mathbb{G}$ and checking whether $\mathbb{G}$ has a cycle can be done in $O(n^2)$ time (the latter using depth-first search). □

At a high level, correctness of the algorithm follows from the observation that if there is a cycle in the directed graph constructed above, then one can construct a complete preference profile $R \in \mathcal{C}(P)$ such that each agent in the cycle strictly prefers the object allocated to the next agent in the cycle under M. Then, trading objects along this cycle would result in a matching M' that, under this completion R, Pareto-dominates M, establishing that M is not NPO.

Computing an NPO Matching, When It Exists

Next, we find a necessary and sufficient condition for a given top-k preference profile P to admit an NPO matching. In the case that an NPO matching exists, we also provide an algorithm to compute one in polynomial time.

Theorem 2. *A top-k preference profile P admits an NPO matching if and only if there is a matching of size at least $n-1$ w.r.t. P. Moreover, there is a polynomial time algorithm to determine whether a top-k preference profile admits an NPO matching, and to compute an NPO matching if it exists.*

Proof. ($\Rightarrow$) Consider a top-k preference profile P that admits an NPO matching M. Suppose for the sake of contradiction that there is no matching w.r.t. P that is of size larger than $n-2$. Then, $|M \cap \text{rev}(P)| \leq n-2$, which implies that there are at least two agents, w.l.o.g. agents a_1 and a_2, who are matched to objects they have not revealed under P. It is easy to see that one can now construct a completion $R \in \mathcal{C}(P)$ such that $M(a_2) \succ_{R_1} M(a_1)$ and $M(a_1) \succ_{R_2} M(a_2)$. Under this completion, exchanging the objects assigned to a_1 and a_2 would Pareto-dominate M, contradicting the fact that M is NPO w.r.t. P.
($\Leftarrow$) Let P be a top-k preference profile such that there exists a matching of size at least $n-1$ w.r.t. P. Construct a weighted bipartite graph $\mathbb{G} = (N \cup O, \text{rev}(P))$, where each edge $(a_i, o_j) \in \text{rev}(P)$ has weight equal to the rank of o_j in P_i. Let M be a matching of min. weight among all matchings of max. cardinality in $\mathbb{G}$. Note that $|M| \geq n-1$.

Let us first consider the case when $|M| = n$. If so, then it is easy to see that M is an NPO matching w.r.t. P. This is because any matching that Pareto-dominates M in any completion $R \in \mathcal{C}(P)$ must have size n w.r.t. P as well and a strictly lower weight than M, which is a contradiction.

Now, consider the case when $|M| = n-1$. So exactly one agent and one object are not matched. Without loss of generality, suppose these are a_n and o_n. Construct a matching $\widehat{M}$ of our problem by adding (a_n, o_n) to M. Thus, $|\widehat{M} \cap \text{rev}(P)| = |M| = n-1$. We wish to show that $\widehat{M}$ is NPO. Suppose for the sake of contradiction that there exists a matching M' which Pareto dominates $\widehat{M}$ w.r.t. some completion $R \in \mathcal{C}(P)$. Hence, M' must assign each agent an object she prefers (under R) at least much as what she receives under $\widehat{M}$, and at least one agent a strictly better (under R) object than what she receives under $\widehat{M}$. Thus, M', when viewed as a matching in $\mathbb{G}$, must have cardinality $n-1$ and weight at most the weight of M. Further, weight of M' is equal to the weight of M only if M' assigns each agent $a_i \neq a_n$ the same object that $\widehat{M}$ assigns. However, any two matchings that differ must differ in the assignment to at least two agents. Hence, we have that M' is a matching in $\mathbb{G}$ with cardinality $n-1$ and weight strictly less than the weight of M, which is a contradiction.

Now, we provide a polynomial-time algorithm to compute an NPO matching, when it exists. From the argument above, it is sufficient to be able to compute a min-weight max-cardinality matching M of the graph $\mathbb{G}$. This can be done in polynomial time using the Hungarian algorithm (Munkres 1957). Then, if $|M| = n$, the algorithm returns M, and if $|M| = n-1$, the algorithm returns $\widehat{M}$ by adding the unmatched agent-object pair. □

Elicitation to Compute an NPO Matching

In this section we turn to the question of elicitation. As mentioned in Section 2, we use the next-best query model, and informally our main goal is to understand how to elicit as little information as possible in order to compute an NPO matching. Towards this end, we first prove, in Theorem 3, a lower bound that shows that any elicitation algorithm has a competitive ratio of $\Omega(\sqrt{n})$. Due to space constraints, the proof appears in the full version (Hosseini et al. 2020).

Theorem 3. *In the next-best query model, if there exists an α-competitive elicitation algorithm for finding a necessarily Pareto optimal matching, then $\alpha \in \Omega(\sqrt{n})$.*

A natural strategy to elicit preferences is to simultaneously ask all the n agents to reveal their next best object until the point at which the designer has enough information to compute an NPO matching. Recall that we can check the latter condition at every step by leveraging our Theorem 2 from Section 3. Although this approach is natural, it is not difficult to see that it results in a competitive ratio of $\Omega(n)$.

Let us describe the high-level idea of our algorithm (Algorithm 1), although the exact details vary slightly in the algorithm in order to optimize the competitive ratio. We follow the aforementioned naïve approach of asking all agents to report their next best object only up to a point where we can match at least $n - \sqrt{n}$ agents to objects they have revealed. At that point, we compute one such matching, focus on the unmatched agents (at most $\sqrt{n}$ of them), and then

```
Input: A profile of top-$k$ preferences $P = (P_1, \dots, P_n)$, a set
    of agents $N$, and a set of objects $O$.
Output: An NPO matching
 1: for each $i \in [n]$, initialize $P_i = \emptyset$
 2: $\mathcal{M} \leftarrow \emptyset$; $k \leftarrow 1$; $s \leftarrow |\mathcal{M}|$
 3: while $s < (n-1)$ do
 4:     if $s \leq (n-1) - \min(\{k-1, \sqrt{n}\})$ then
 5:         for each $i \in [n]$ do
 6:             $o \leftarrow \mathcal{Q}(a_i, |P_i|+1)$
 7:             update $P_i$ by adding object $o$ at position $|P_i|+1$
 8:         end for
 9:     else if $s > (n-1) - \min(\{k-1, \sqrt{n}\})$ then
10:         $U \leftarrow$ the set of unmatched agents in $\mathcal{M}$
11:         for each $i \in U$ do
12:             $o \leftarrow \mathcal{Q}(a_i, |P_i|+1)$
13:             update $P_i$ by adding object $o$ at position $|P_i|+1$
14:         end for
15:     end if
16:     $k \leftarrow k+1$
17:     $\mathcal{M} \leftarrow$ max. matching in $\mathbb{G}(N \cup O, \text{rev}(P))$
18:     $s \leftarrow |\mathcal{M}|$
19: end while
20: use algorithm described in Theorem 2 to return a matching
    that is NPO w.r.t. $P = (P_1, \cdots, P_n)$
```

Algorithm 1: A $2(\sqrt{n}+1)$-competitive elicitation algorithm

just ask these agents for their next best object until there is a matching of size at least $n-1$, which, by Theorem 2, is a sufficient condition for the existence of an NPO matching. Once we know an NPO matching exists, we use the polynomial time algorithm from Theorem 2 to find one. This strategy, with the details optimized, results in an improved competitive ratio of $2(\sqrt{n}+1)$, which, given Theorem 3, is asymptotically optimal.

Theorem 4. *Algorithm 1 is a $2(\sqrt{n}+1)$-competitive elicitation algorithm in the next-best query model for computing a necessarily Pareto optimal matching.*

In order to prove this result, we first introduce the following notation. We maintain a graph $\mathbb{G} = (N \cup O, \text{rev}(P))$, where P denotes the top-k preference profile elicited so far. Let s_j denote the size of the maximum matching in this graph during the j-th iteration of line 3. Next, let m denote the number of times lines 4–8 are run in Algorithm 1. Note that $m \geq 1$ (since $s_1 = 0$ and hence lines 4–8 are executed at least during the first iteration), and that the total number of iterations of the algorithm is $m+1$ (since the 'else if', i.e., lines 9–15 is executed just once). Also, note that for all $j \in [m]$, s_j is also the size of the maximum matching when all agents have revealed their top $(j-1)$ objects in Algorithm 1. Finally, for $j \in [m]$, let X_j denote the number of agents who are asked at least j queries by the optimal algorithm. Now, we have the following claim, whose proof appears in the full version (Hosseini et al. 2020).

Claim 1. *For every $j \in [m]$, $X_j \geq (n - s_j - 1)$.*

Equipped with the notations and the claim above, we can now prove our theorem.

Proof of Theorem 4. Let ALG denote the number of queries asked in Algorithm 1 and m be as defined above. Since the algorithm asks n queries during each time lines 4–8 is run and it asks at most $(n - s_{m+1}) \cdot (n-m)$ queries during the execution of lines 9–15 (the first term is maximum number of agents in set U in line 10 and the second arises from the fact that all these agents have revealed their top m preferences), we have that

$$\begin{aligned} \text{ALG} &\leq n \cdot m + (n - s_{m+1}) \cdot (n-m) \\ &< n \cdot m + (\min(\{m, \sqrt{n}\}) + 1) \cdot (n-m), \qquad (1) \end{aligned}$$

where the second step follows from the fact that $(n - s_{m+1}) < (1 + \min(\{m, \sqrt{n}\}))$ (see line 9 in Algorithm 1).

Next, let OPT denote the number of queries asked by the optimal algorithm. From Claim 1 we know that X_j, which is the number of agents who are asked at least j queries by the optimal algorithm, is at least $(n - s_j - 1)$. Therefore,

$$\begin{aligned} \text{OPT} &\geq m \cdot X_m + \sum_{j=1}^{m-1} (m-j) \cdot (X_{m-j} - X_{m-j+1}) \\ &= X_m + X_{m-1} + \cdots + X_1 \\ &\geq n - 1 + \sum_{j=2}^{m} (n - s_j - 1) \\ &\qquad \text{(using Claim 1 and the fact that } s_1 = 0\text{)} \\ &\geq n - 1 + (m-1) \cdot (n - s_m - 1) \\ &\qquad \text{(since } \forall j \in [m], s_j \leq s_{j+1}\text{)} \\ &\geq n - 1 + (m-1) \cdot \min(\{m-1, \sqrt{n}\}), \qquad (2) \end{aligned}$$

where the final inequality follows from the fact that $(n - 1 - s_m) \geq \min(\{m-1, \sqrt{n}\})$ (see line 4 in Algorithm 1). Therefore, using (1) and (2), we have

$$\begin{aligned} \frac{\text{ALG}}{\text{OPT}} &\leq \frac{n \cdot m + (\min(\{m, \sqrt{n}\}) + 1) \cdot (n-m)}{n - 1 + (m-1) \cdot \min(\{m-1, \sqrt{n}\})} \\ &\leq 2(\min(\{m, \sqrt{n}\}) + 1), \end{aligned}$$

where the last inequality follows by using $n \geq 2$, $m \geq 1$, and by considering the cases $m \leq \sqrt{n}$ and $m > \sqrt{n}$. □

4 Necessarily Rank-Maximal Matchings

In the previous section, we devised algorithms for computing an NPO matching when one exists and when given a top-k preference profile, and for eliciting a small amount of information to determine such matchings. Here, we focus on rank-maximality, which is a stronger notion of economic efficiency than Pareto optimality and widely used in real-world applications (see, e.g., (Garg et al. 2010; Abraham et al. 2006; Abraham 2009)). To illustrate this notion, consider the top-k preference profile P with three agents, where $P_1 = o_1 \succ o_2 \succ o_3$, $P_2 = o_1 \succ o_2$ and $P_3 = o_1$. The matching $M = \{(a_1, o_3), (a_2, o_2), (a_3, o_1)\}$ is NPO, but not NRM. To see this, consider the completion R of P, where $R_1 = o_1 \succ o_2 \succ o_3$, $R_2 = o_1 \succ o_2 \succ o_3$, and $R_3 = o_1 \succ o_3 \succ o_2$. While M has signature $(1, 1, 1)$ under R, matching $M' = \{(a_1, o_1), (a_2, o_2), (a_3, o_3)\}$ has a better signature $(1, 2, 0)$ under R. In fact, it is easy to verify that M' is NRM w.r.t. P.

We begin, in Section 4, by presenting an algorithm that determines whether a given matching is necessarily rank-maximal (NRM) w.r.t. a given top-k preference profile. In Section 4, we build upon this algorithm to design another algorithm that decides if a given top-k preference profile admits an NRM matching, and computes one if it does. Finally, we turn to the elicitation question, where we provide a lower bound on the competitive ratio of any elicitation algorithm that returns an NRM matching. We first introduce necessary additional definitions and notations.

Definition 5 (rank-maximality w.r.t. top-k preferences)**.** Given a top-k preference profile P and a matching M, let the *signature* of M w.r.t. P, denoted $\text{sig}_P(M)$, be the tuple $(x_1, \dots, x_n)$, where x_ℓ is the number of $(a_i, o_j) \in M \cap \text{rev}(P)$ such that o_j is the ℓ^{th} most preferred choice of a_i. We say that $(x_1, \dots, x_n) \succ_{\text{sig}} (x'_1, \dots, x'_n)$ if there exists a $h \in [n]$ such that $x_h > x'_h$ and $x_i = x'_i$ for all $i < h$; this is the lexicographic comparison of signatures. A matching M is rank-maximal w.r.t. P if $\text{sig}_P(M)$ is maximum under $\succ_{\text{sig}}$. In plain words, a matching is rank-maximal w.r.t. partial preference profile P if it maximizes the number of agents matched to their top choice, subject to that the number of agents matched to their second choice, and so on.

Definition 6 (Extended signature)**.** Following Definition 2, we define the *extended signature* of a matching M w.r.t. P, denoted $\text{extsig}_P(M)$, to be the tuple $(x_1, \dots, x_n)$, where, for all $i \in [n-1]$, x_i is defined as in $\text{sig}_P(M)$, while x_n counts not only the number of agents matched to their n^{th} choice revealed in P, but also the number of agents matched to an object they did not reveal in P.

Definition 7 (Optimal signature)**.** Given a top-k preference profile P over a subset of agents $S \subseteq N$ and a subset of objects $T \subseteq O$, as well as a set of "forbidden" pairs $F \subseteq S \times T$, let $\mathcal{M}(S, T, F)$ denote the set of all matchings M that match agents in S to objects in T and satisfy $M \cap F = \emptyset$. We define $\text{sig}_P^{\text{opt}}(S, T, F) = \sup_{M \in \mathcal{M}(S,T,F), R \in \mathcal{C}(P)} \text{sig}_R(M)$. Informally, this is the *best possible signature* that any matching can achieve under any completion of P while avoiding pairings from F. We use $\arg \text{sig}_P^{\text{opt}}(S, T, F)$ to denote the matching $M \in \mathcal{M}(S, T, F)$ that obtains this best signature under some completion $R \in \mathcal{C}(P)$. Note that sometimes we use $\text{sig}_P^{\text{opt}}$ to denote $\text{sig}_P^{\text{opt}}(N, O, \emptyset)$.

Determining Whether a Matching is NRM

At a high level, our algorithm to determine if a given matching M is NRM under a given top-k preference profile P works as follows. It leverages the fact that given any $S \subseteq N$, $T \subseteq O$, and $F \subseteq S \times T$, we can efficiently compute $\text{sig}_P^{\text{opt}}(S, T, F)$. Algorithm 2 formally describes how to do this; the proof of its correctness is in the full version.

Given this, our algorithm first checks if M has size at least $n-1$ w.r.t. P; this is necessary for M to even be NPO w.r.t. P, as the proof of Theorem 2 shows. If M has size n w.r.t. P, then we argue that its signature w.r.t. P (which is the same as its signature w.r.t. any completion R of P) must be $\text{sig}_P^{\text{opt}}$. On the other hand, suppose M has size $n-1$ w.r.t. P, with $(a_i, o_j) \in M \cap \text{unrev}(P)$ being the pair where an agent is matched to an object she did not reveal under P. Then, we argue that two conditions need to be met: (a) to ensure that another matching M' with $(a_i, o_j) \in M'$ cannot have a better signature under any completion R, we require that the signature of M w.r.t. P be at least as good as $\text{sig}_P^{\text{opt}}(N \setminus \{a_i\}, O \setminus \{o_j\}, \emptyset)$, and (b) to ensure that another matching M' with $(a_i, o_j) \notin M'$ cannot have a better signature under any completion R, we require that the extended signature of M w.r.t. P be at least as good as $\text{sig}_P^{\text{opt}}(N, O, \{(a_i, o_j)\})$. Algorithm 3 formalizes this and the proof of its correctness (claimed below) appears in the full version.

```
Input: A top-$k$ preference profile $P$ over a set of agents $S \subseteq N$ and a set of objects $T \subseteq O$, and $F \subseteq S \times T$.
Output: $\text{sig}_P^{\text{opt}}(S, T, F)$ and $\arg \text{sig}_P^{\text{opt}}(S, T, F)$
For all $a_i \in S$, let $k_i \leftarrow |P_i|$
$P' \leftarrow$ weak order profile which matches $P$ on the preferences revealed under $P$, and for every $(a_i, o_j) \in \text{unrev}(P)$, has object $o_j$ at position $k_i + 1$ in $P'_i$
Create a bipartite graph $\mathbb{G} = (S \cup T, E)$, where $E = S \times T \setminus F$, and each edge $(a_i, o_j) \in E$ is labelled with $\ell$ if $o_j$ appears in position $\ell$ in $P'_i$.
$M \leftarrow$ compute a rank-maximal matching of $\mathbb{G}$ using the algorithm by Irving et al. (2006)
return $\text{sig}_{P'}(M)$ and $M$
```

Algorithm 2: Algorithm to compute $\text{sig}_P^{\text{opt}}(S, T, F)$ and $\arg \text{sig}_P^{\text{opt}}(S, T, F)$.

```
Input: A top-$k$ preference profile $P$, a set of agents $N$, a set of objects $O$, and a matching $M$.
Output: Is $M$ NRM w.r.t. $P$?
$c \leftarrow |M \cap \text{rev}(P)|$.
if $c = n$ and $\text{sig}_P(M) = \text{sig}_P^{\text{opt}}$ then
    return YES
else if $c = n - 1$ then
    Pick $(a_i, o_j) \in M \cap \text{unrev}(P)$ ▷ this is unique since $M$ has cardinality $n-1$ w.r.t. $P$.
    if $\text{sig}_P(M) \succeq_{\text{sig}} \text{sig}_P^{\text{opt}}(N \setminus \{a_i\}, O \setminus \{o_j\}, \emptyset)$ and $\text{extsig}_P(M) \succeq_{\text{sig}} \text{sig}_P^{\text{opt}}(N, O, \{(a_i, o_j)\})$ then
        return YES
    end if
end if
return NO
```

Algorithm 3: Algorithm to check if a given matching is NRM given a top-k preference profile.

Theorem 5. *Given a top-k preference profile P and a matching M, there exists a polynomial-time algorithm that determines if M is NRM w.r.t. P.*

Computing an NRM matching, When It Exists

The results from Section 4 are key to devising an algorithm that determines if a given top-k preference profile admits an NRM matching and computes one if it does. Specifically, note again that a matching that is NRM w.r.t. a top-k preference profile P must have size at least $n-1$ w.r.t. P. If there is an NRM matching of size n w.r.t. P, then we argue that it

```
Input: A top-$k$ preference profile $P$, agents $N$, and objects $O$.
Output: Returns an NRM matching if $P$ admits one, and returns NO otherwise.
 1: $M' \leftarrow$ rank-maximal matching w.r.t. $P$
 2: if $|M' \cap rev(P)| = n$ and $\text{sig}_P(M) = \text{sig}_P^{\text{opt}}$ then
 3:     return $M'$
 4: else
 5:     for $(a_i, o_j) \in \text{unrev}(P)$ do
 6:         $X \leftarrow$ rank-maximal matching of $N \setminus \{a_i\}$ to $O \setminus \{o_j\}$ w.r.t. $P$
 7:         $M' \leftarrow X \cup \{(a_i, o_j)\}$
 8:         if $M'$ is NRM w.r.t. $P$, then return $M'$
 9:     end for
10: end if
11: return NO
```

Algorithm 4: Algorithm to check if an NRM matching exists given a top-k preference profile.

must be rank-maximal w.r.t. P and have signature $\text{sig}_P^{\text{opt}}$. We can check this efficienty using the results from Section 4. If there is an NRM matching of size $n - 1$ w.r.t. P with $(a_i, o_j) \in M \cap \text{unrev}(P)$, then we argue that $M \setminus \{(a_i, o_j)\}$ must be a rank-maximal matching of $N \setminus \{a_i\}$ to $O \setminus \{o_j\}$ w.r.t. P. By iterating over all agent-object pairs (a_i, o_j), we can again check this efficiently. Algorithm 4 formalizes this and the next result proves its correctness.

Theorem 6. *Given a top-k preference profile P, there exists a polynomial-time algorithm that returns a necessarily rank-maximal matching if one exists, and returns NO otherwise.*

Proof. We want to show that P admits an NRM matching if and only if Algorithm 4 returns a matching (i.e. does not return NO). Moreover, in case that the algorithm returns a matching, we want to show that it is in fact NRM w.r.t. P.

($\Rightarrow$) Let M be an NRM matching w.r.t. P. Then, it must be NPO w.r.t. P. Hence, as we show in the proof of Theorem 2, M must have size at least $n - 1$ w.r.t. P. So, we again have the following two cases. Either M is a perfect matching w.r.t. P, which we refer to as Case (a) below, or M is of size $n - 1$ w.r.t. P, which we refer to as Case (b) below.

Case (a). Observe that $\text{sig}_P(M) = \text{sig}_R(M)$ for all $R \in \mathcal{C}(P)$ since M is a perfect matching w.r.t. P. Second, note that $\text{sig}_P(M) = \text{sig}_P^{\text{opt}}$, since otherwise it follows from the definition of $\text{sig}_P^{\text{opt}}$ that there is a matching M' and a completion $R \in \mathcal{C}(P)$ such that $\text{sig}_R(M') \succ_{\text{sig}} \text{sig}_R(M) = \text{sig}_P(M)$, which in turn contradicts the fact that M is NRM w.r.t. P. Finally, it is easy to see that since M has size n w.r.t. P and it is NRM w.r.t. P, it has to be rank-maximal w.r.t. to P. Combining these observations, we have that any matching M' that is a rank-maximal matching w.r.t. P must have size n and the same signature (under P) as $\text{sig}_P^{\text{opt}}$. Hence, the check in line 2 will succeed, and the algorithm will return matching M', which is NRM w.r.t. P.

Case (b). Let $(a_i, o_j) \in M \cap \text{unrev}(P)$. As in Case (a), we can argue that $M \setminus \{(a_i, o_j)\}$ is NRM w.r.t. P among all matchings that match $N \setminus \{a_i\}$ to $O \setminus \{o_j\}$. Let X be any matching of $N \setminus \{a_i\}$ to $O \setminus \{o_j\}$ that is rank-maximal w.r.t. P. Then, this implies that $X \cup \{(a_i, o_j)\}$ is also an NRM matching. Observe that this is precisely the condition that is checked in line 8 of Algorithm 4. Hence, the algorithm returns an NRM matching w.r.t. P in this case too.

($\Leftarrow$) Suppose P does not have an NRM matching. Note that Algorithm 4 returns a matching only under two cases. First is when there is a matching M' such that M' is of size n w.r.t. P and has the same signature as $\text{sig}_P^{\text{opt}}$. Note that if such a matching exists, then it is easy to see from the definition of $\text{sig}_P^{\text{opt}}$ and the fact that M' is a perfect matching w.r.t. P that it has to be NRM w.r.t. P, which in turn contradicts our assumption that P does not have an NRM.

The other case is when Algorithm 4 returns a matching in line 8, and here, the proof trivially follows from Theorem 5, which proves the correctness of Algorithm 3.

Finally, one can see that Algorithm 4 runs in polynomial time since Algorithms 2 and 3 run in polynomial time. □

Elicitation to Compute an NRM Matching

We are now ready to consider the question of eliciting information to compute an NRM matching. Below we provide a lower bound on the competitive ratio of any elicitation algorithm (the proof is in the full version (Hosseini et al. 2020)).

Theorem 7. *In the next-best query model, for $n \geq 2$, if there exists an α-competitive elicitation algorithm for finding a necessarily rank-maximal matching, then $\alpha \geq \frac{4}{3} - \frac{2}{n}$, if n is even, and $\alpha \geq \frac{4}{3} - \frac{20}{9n-3}$, if n is odd.*

Note that our lower bound converges to $\frac{4}{3}$ as n goes to infinity. So, a natural question is whether one can design an online algorithm to compute an NRM matching that has constant competitive ratio. We conjecture that this should be possible, but leave it as an open problem for future work.

5 Discussion

As discussed in Section 4, the most immediate open question that stems from our work is to design an online elicitation algorithm to compute an NRM matching, and analyze its competitive ratio. In particular, we believe that there exists an algorithm with a constant competitive ratio.

Note that our online algorithm to compute an NPO matching from Section 3 is a constructive procedure, and uses our result that the existence of an NPO matching can be reduced to a simple analytical condition. Given that such a simple condition seems out of reach for the case of NRM, it may be interesting to explore a different approach to designing an online algorithm to compute an NRM matching. For example, one can train a machine learning model to select which agent to query next as a function of the preferences elicited so far. At each step, we can first use this algorithm to make one more query, and then call our efficient algorithm to check if the elicited preference reveal an NRM matching. It would be interesting to study if this approach can lead to a low competitive ratio, at least in practice.

More broadly, the literature on learning a desirable matching of agents to objects from partial preferences or historical data is still in its infancy, and many interesting directions, such as studying other models of partial preferences or other models of querying agents, remain unexplored.

Acknowledgments

Hadi Hosseini acknowledges support from NSF grant #1850076. Nisarg Shah was partially supported by an NSERC Discovery grant. We thank Lirong Xia and the anonymous reviewers for their very helpful comments and suggestions.

References

Abdulkadiroğlu, A.; and Sönmez, T. 1998. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica* 66(3): 689–701.

Abdulkadiroğlu, A.; and Sönmez, T. 1999. House allocation with existing tenants. *Journal of Economic Theory* 88(2): 233–260.

Abraham, D.; Chen, N.; Kumar, V.; and Mirrokni, V. S. 2006. Assignment problems in rental markets. In *Proceedings of the Second International Workshop on Internet and Network Economics (WINE)*, 198–213. Springer.

Abraham, D. J. 2009. *Matching markets: Design and analysis*. Ph.D. thesis, University of Glasgow.

Abraham, D. J.; Cechlárová, K.; Manlove, D. F.; and Mehlhorn, K. 2004. Pareto optimality in house allocation problems. In *Proceedings of the Fifteenth International Symposium on Algorithms and Computation (ISAAC)*, 3–15.

Aziz, H.; Biro, P.; de Haan, R.; and Rastegari, B. 2019a. Pareto optimal allocation under compact uncertain preferences. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 1740–1747.

Aziz, H.; Biró, P.; Lang, J.; Lesca, J.; and Monnot, J. 2019b. Efficient reallocation under additive and responsive preferences. *Theoretical Computer Science* 790: 1–15.

Aziz, H.; Gaspers, S.; Mackenzie, S.; and Walsh, T. 2015. Fair assignment of indivisible objects under ordinal preferences. *Artificial Intelligence* 227: 71–92.

Aziz, H.; Walsh, T.; and Xia, L. 2015. Possible and necessary allocations via sequential mechanisms. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 468–474.

Bogomolnaia, A.; and Moulin, H. 2001. A new solution to the random assignment problem. *Journal of Economic theory* 100(2): 295–328.

Borodin, A.; and El-Yaniv, R. 2005. *Online computation and competitive analysis*. Cambridge University Press.

Bouveret, S.; Endriss, U.; and Lang, J. 2010. Fair Division under Ordinal Preferences: Computing Envy-Free Allocations of Indivisible Goods. In *Proceedings of Nineteenth European Conference on Artificial Intelligence (ECAI)*, volume 215, 387–392.

Che, Y.-K.; and Kojima, F. 2010. Asymptotic equivalence of probabilistic serial and random priority mechanisms. *Econometrica* 78(5): 1625–1672.

Cirillo, R. 2012. *The Economics of Vilfredo Pareto*. Routledge.

Drummond, J.; and Boutilier, C. 2013. Elicitation and approximately stable matching with partial preferences. In *Proceedings of the Twenty-Third international Joint conference on Artificial Intelligence (IJCAI)*, 97–105. AAAI Press.

Garg, N.; Kavitha, T.; Kumar, A.; Mehlhorn, K.; and Mestre, J. 2010. Assigning papers to referees. *Algorithmica* 58(1): 119–136.

Gaspers, S.; Naroditskiy, V.; Narodytska, N.; and Walsh, T. 2014. Possible and necessary winner problem in social polls. In *Proceedings of the Thirteenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 613–620.

Hosseini, H.; Larson, K.; and Cohen, R. 2018. Investigating the characteristics of one-sided matching mechanisms under various preferences and risk attitudes. *Autonomous Agents and Multi-Agent Systems* 32(4): 534–567.

Hosseini, H.; Menon, V.; Shah, N.; and Sikdar, S. 2020. Necessarily Optimal One-Sided Matching. *arXiv preprint arXiv:2007.09079* .

Hylland, A.; and Zeckhauser, R. 1979. The efficient allocation of individuals to positions. *Journal of Political Economy* 87(2): 293–314.

Irving, R. W. 2003. Greedy matchings. *Technical Report TR-2003-136, University of Glasgow* .

Irving, R. W.; Kavitha, T.; Mehlhorn, K.; Michail, D.; and Paluch, K. E. 2006. Rank-maximal matchings. *ACM Transactions on Algorithms (TALG)* 2(4): 602–610.

Liu, Q.; Mailath, G. J.; Postlewaite, A.; and Samuelson, L. 2014. Stable matching with incomplete information. *Econometrica* 82(2): 541–587.

Manlove, D. 2013. *Algorithmics of matching under preferences*, volume 2. World Scientific.

Moulin, H. 2004. *Fair division and collective welfare*. MIT press.

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 5(1): 32–38.

Rastegari, B.; Condon, A.; Immorlica, N.; and Leyton-Brown, K. 2013. Two-sided matching with partial information. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce (EC)*, 733–750.

Shapley, L.; and Scarf, H. 1974. On cores and indivisibility. *Journal of Mathematical Economics* 1(1): 23–37.

Xia, L.; and Conitzer, V. 2011. Determining Possible and Necessary Winners Given Partial Orders. *Journal of Artificial Intelligence Research (JAIR)* 41: 25–67.