

A Few Queries Go a Long Way: Information-Distortion Tradeoffs in Matching

Georgios Amanatidis

GEORGIOS.AMANATIDIS@ESSEX.AC.UK

*Department of Mathematical Sciences
University of Essex
Colchester, CO4 3SQ, United Kingdom*

Georgios Birmbas

GEORGE.BIRBAS@DIAG.UNIROMA1.IT

*Department of Computer, Control and Management Engineering
Sapienza University of Rome
Piazzale Aldo Moro, 5, 00185 Roma RM, Italy*

Aris Filos-Ratsikas

ARIS.FILOS-RATSIKAS@LIVERPOOL.AC.UK

*Department of Computer Science
University of Liverpool
Liverpool, L69 3BX, United Kingdom*

Alexandros A. Voudouris

ALEXANDROS.VOUDOURIS@ESSEX.AC.UK

*School of Computer Science and Electronic Engineering
University of Essex
Colchester, CO4 3SQ, United Kingdom*

Abstract

We consider the *One-Sided Matching* problem, where n agents have preferences over n items, and these preferences are induced by underlying cardinal valuation functions. The goal is to match every agent to a single item so as to maximize the *social welfare*. Most of the related literature, however, assumes that the values of the agents are not a priori known, and only access to the ordinal preferences of the agents over the items is provided. Consequently, this incomplete information leads to loss of efficiency, which is measured by the notion of *distortion*. In this paper, we further assume that the agents can answer a small number of *queries*, allowing us partial access to their values. We study the interplay between elicited cardinal information (measured by the number of queries per agent) and distortion for One-Sided Matching, as well as a wide range of well-studied related problems. Qualitatively, our results show that with a limited number of queries, it is possible to obtain significant improvements over the classic setting, where only access to ordinal information is given.

1. Introduction

In the *One-Sided Matching* problem (often referred to as the *house allocation* problem), n agents have *preferences* over a set of n items, and the goal is to find an allocation in which every agent receives a single item, while maximizing some objective. Typically, as well as in this paper, this objective is the (utilitarian) social welfare, i.e., the total utility of the agents. Since its introduction by Hylland and Zeckhauser (1979), this has been one of the most fundamental problems in the literature of economics (e.g., see Bogomolnaia & Moulin,

2001; Svensson, 1999), and has also been extensively studied in computational social choice (e.g., see Klaus et al., 2016).

The classic work on the problem (including Hylland and Zeckhauser’s seminal paper) assumes that the preferences of the agents are captured by cardinal valuation functions, assigning numerical values to the different items; these can be interpreted as their *von Neuman-Morgenstern utilities* (Von Neumann & Morgenstern, 1947). From a more algorithmic viewpoint, one can envision a weighted complete bipartite graph (with agents and items forming the two sides of the partition), where the weights of the edges are given by these values. Crucially, most of the related literature assumes that the designer only has access to the preference rankings of the agents over the items (i.e., the *ordinal preferences*) induced by the underlying values, but not to the values themselves.¹ This is motivated by the fact that it is fairly standard to ask the agents to simply order the items, while it is arguably much more demanding to require them to specify exact numerical values for all items.

This begs the following natural question: *What is the effect of this limited information on the goals of the algorithm designer?* In 2006, Procaccia and Rosenschein defined the notion of *distortion* to measure precisely this effect, when the goal is to maximize the social welfare. Their original research agenda was put forward for settings in general social choice (also referred to as *voting*), but has since then flourished to capture several different scenarios, including One-Sided Matching (Anshelevich et al., 2021). For the latter problem, Filos-Ratsikas et al. (2014), showed that the best possible distortion achieved by any ordinal algorithm is $\Theta(\sqrt{n})$, even using randomization, and even if the valuations are normalized. For deterministic algorithms, we show that the corresponding bound is $\Theta(n^2)$ (Theorem 1).

While the aforementioned bounds establish a stark impossibility when one has access only to ordinal information, they do not rule out the prospect of good approximations when it is possible to elicit *some* cardinal information. Indeed, the cognitive burden of eliciting cardinal values in the literature has mostly been considered in the two extremes; either full cardinal information or not at all. Conceivably though, if the agents are needed to come up with only *a few* cardinal values, the elicitation process would not be very demanding, while it could potentially have wondrous effects on the social welfare. This approach was advocated recently by Amanatidis et al. (2021), who proposed to study the *tradeoffs* between the number of cardinal *value queries* per agent and distortion. For the general social choice setting of Procaccia and Rosenschein (2006), Amanatidis et al. (2021) actually showed that with a limited number of such queries, one can significantly improve upon the existing strong impossibilities (Boutilier et al., 2015; Caragiannis et al., 2017). Motivated by the success of this approach for general social choice settings, we extend this research agenda and aim to answer the following question for One-Sided Matching:

What are the best possible information-distortion tradeoffs in One-Sided Matching? Can we achieve significant improvements over the case of only ordinal preferences, by making only a few cardinal value queries per agent?

1. The pseudo-market mechanism of Hylland and Zeckhauser (1979) is a notable exception.

1.1 Our Contribution

We consider the One-Sided Matching problem with the goal of maximizing the social welfare under limited information. We adopt the standard assumption in the related literature that the agents provide as input their ordinal preferences over the items, and that these are induced by their cardinal valuation functions. Following the agenda put forward by Amanatidis et al. (2021), we also assume implicit access to the numerical values of the agents via *value queries*; we may ask for an agent i and an item j , and obtain the agent's value, $v_i(j)$, for that item.

We measure the performance of an algorithm by the standard notion of *distortion*, and our goal is to explore the tradeoffs between distortion and the number of queries we need per agent. As the two extremes, we note that if we use n queries per agent, we recover the complete cardinal valuation profile and thus the distortion is 1, whereas if we use 0 queries, i.e., we use only the ordinal information, the best possible distortion is $\Theta(n^2)$ (see Theorem 1). The latter bound holds even if we consider valuation functions that satisfy the *unit-sum* normalization, i.e., the sum of the values of each agent for all the items is 1. As we mentioned earlier, even when allowing randomization, the best possible distortion is still quite large ($\Theta(\sqrt{n})$; Filos-Ratsikas et al., 2014) without employing any value queries. In this work, we only consider deterministic algorithms, and leave the study of randomized algorithms for future work.

We provide the following results:

- In Section 3, we present an algorithm parametrized by λ , which achieves distortion $O(n^{1/(\lambda+1)})$ by making $O(\lambda \log n)$ queries per agent. In particular, by setting $\lambda = 1$ and $\lambda = O(\log n)$ we achieve respectively
 - distortion $O(\sqrt{n})$ using $O(\log n)$ queries per agent;
 - *constant* distortion using $O(\log^2 n)$ queries per agent.

The algorithm is inspired by a conceptually similar idea presented by Amanatidis et al. (2021) for the social choice setting. In Section 6 we adapt our algorithm to provide analogous information-distortion tradeoffs for a wide range of well-studied optimization problems, including *Two-Sided Matching*, *General Graph Matching* and the *Clearing Kidney Exchanges*.

- Next, still in Section 3, motivated by the analysis of the class of algorithms mentioned above as well as our lower bounds in Section 4, we consider a class of instances (coined *k-well-structured instances*) that captures the case where the agents (roughly) agree on the ranking of the items. We present a simple algorithm achieving distortion $O(k \cdot n^{1/k})$ by making only k queries per agent for these instances.
- In Section 4 we show a lower bound of $\Omega(n^{1/k}/k)$ on the distortion of any algorithm that makes k queries per agent. An immediate consequence of this bound is that it is impossible to achieve constant distortion without asking almost $\log n$ queries! When k is a constant, our aforementioned results on *k-well-structured instances* also establish the tightness of our construction, since the proof uses instances of this type. Furthermore, using a construction which exploits the same ordinal but different cardinal

| | | | | | | | |
|--------------------------------|----------------------------|-----------------------------------------------|---------------------------------------------|---------------------------------------------------------|--------------------------------|--------------------------|--------------------------|
| unit-sum valuations | $\Theta(n^2)$ | $\tilde{O}(n^{2/3})$ $\Omega(n^{1/3})$ | $\Omega(n^{1/(k+1)})$ | $\Omega(\frac{n^{1/(k+1)}}{k})$ | | | |
| unrestricted valuations | unbounded | $O(\sqrt{n})$ [2-WS] $\Omega(\sqrt{n})$ | $O(n^{1/k})$ [k-WS] $\Omega(n^{1/k})$ | $O(k n^{1/k})$ [k-WS] $\Omega(\frac{n^{1/k}}{k})$ | $O(n^{1/(\lambda+1)})$ | $O(1)$ | 1 |
| | ordinal info, 0 queries | 2 queries | k queries ($k = \Theta(1)$) | k queries ($k = \omega(1)$) | $O(\lambda \log n)$ queries | $O(\log^2 n)$ queries | full info n queries |

Figure 1: An overview of our results. Some of the distortion guarantees hold for k -well-structured instances; this is indicated in brackets next to the bound. The upper bounds that hold for unrestricted valuations also obviously hold for unit-sum valuations. All of the lower bounds hold even for instances with the same ordinal preferences for all agents.

information, we show that even under the stronger assumption of unit-sum normalization, the distortion cannot be better than $\Omega(n^{1/(k+1)}/k)$ with k queries per agent.

- In Section 5 we present our main algorithmic result for unit-sum valuations, namely a novel algorithm which achieves distortion $O(n^{2/3}\sqrt{\log n})$ using only *two* queries per agent.

Our results are summarized in Figure 1. We note that our upper bounds for unrestricted valuations are robust to “errors” in the responses to the queries (albeit not stated this way for the sake of simplicity). As long as the reported values are within a constant multiplicative factor from the true values, qualitatively there is no change in any of our bounds.

TECHNICAL OVERVIEW

Our parametrized class of algorithms developed in Section 3 is based on the following idea. First, every agent is queried for their favorite item. Then, for each agent, the algorithm partitions the items into sets so that the value of the agent for all items in a set is lower-bounded by a carefully defined quantity; these sets are constructed via a sequence of binary search subroutines. Finally, the algorithm outputs a matching that maximizes the social welfare, with respect to the “simulated” values obtained via this process.

A similar idea was proposed by Amanatidis et al. (2021) for the social choice setting. We remark, however, that if one translates their result directly to our setting (by interpreting matchings as *alternatives* and running their algorithm), it yields inconsequential distortion bounds, as well as an exponential-time running time for the algorithm. To obtain meaningful bounds, the key is to adopt the *principle* of their approach rather than the exact solution proposed. As a matter of fact, in Section 6, we show that the same principle can be further refined and applied to a plethora of other combinatorial optimization problems on graphs with additive objectives. In particular, we show bounds of similar flavor for several well-studied problems, such as:

- Two-Sided (Perfect) Matching (i.e., a (perfect) matching on a bipartite graph with agents on each side);
- General Graph Matching (i.e., a matching on a general graph with agents being the vertices);
- General Resource Allocation (i.e., the allocation of m items to n agents under various constraints);
- Max k -Sum Clustering (i.e., a generalization of matchings on graphs, see the paper of Anshelevich & Sekar, 2016a);
- Clearing Kidney ℓ -Exchanges (i.e., a cycle cover with restricted cycle length, see the paper of Abraham et al., 2007).

With regard to our results for k -well-structured instances in Section 3, we note that we obtain a notable improvement over the tradeoffs achieved by the aforementioned approach. For instance, a distortion of $O(\sqrt{n})$ is achievable here using only *two* queries! To provide some intuition, one can think of k -well-structured instances as capturing cases for which there is a general agreement on the set of the “most valuable items”, although the agents might rank the items in this set in different ways. For example, most researchers in Artificial Intelligence would agree on the top 5 publication venues, although they might rank those 5 venues differently. The parameter k captures the different “levels” of agreement.

An interesting class of instances that are k -well-structured for every k is that of instances where all the agents have the *same ranking* over all items. These instances are important because intuitively, they highlight the challenge of social welfare maximization under ordinal information. How is an algorithm in that case supposed to distinguish between pairs of high and low value? Perhaps somewhat surprisingly, it turns out that such instances are more amenable to handling via a smaller number of queries. In fact, the related literature has been concerned in the past with this type of instances; e.g., Filos-Ratsikas et al. (2014) used such instances in their lower bound constructions and referred to them as *ordered* instances. Plaut and Roughgarden (2020) and Barman and Krishnamurthy (2020) also considered such instances in the context of fair division of indivisible items. We also use such instances for our lower bound constructions in Section 4; in that sense, we do not only provide improved upper bounds for an interesting class of instances but we also show the tightness of the analysis for our lower bound constructions.

We remark that the results of Sections 3 and 6 do not require any normalization assumptions. For agents with unit-sum normalized valuations, in Section 5, we present an algorithm which achieves a distortion of $O(n^{2/3}\sqrt{\log n})$ using only two queries per agent. The algorithm is adaptive, and uses the second query differently, depending on the maximum value that it sees after querying all agents at the first position. While this result is not tight based on our lower bound of $\Omega(n^{1/3})$ for this case, we consider it as one of the highlights of this work. In particular, it shows that sublinear distortion is possible even with a deterministic algorithm using a constant number of queries per agent.

1.2 Related Work

The One-Sided Matching problem in the context of agents with preferences over items was firstly introduced by Hylland and Zeckhauser (1979). The classic literature in economics (e.g., see Bogomolnaia & Moulin, 2001; Svensson, 1999 and references therein) was mostly concerned with axiomatic properties, and has proposed several solutions and impossibilities; see the surveys of Sönmez and Ünver (2011) and Abdulkadiroglu and Sönmez (2013) for more information.

The effects of limited information on the social welfare objective were studied most notably in the work of Filos-Ratsikas et al. (2014) mentioned earlier. Further, a line of work Anshelevich & Sekar, 2016a, 2016b; Anshelevich & Zhu, 2017; Abramowitz & Anshelevich, 2018 studied related settings on graphs, and showed distortion bounds for matching problems and their generalizations. A crucial difference from our work is that they consider *symmetric* weights on the edges of the graph, which corresponds to cases where agents are paired with other agents (e.g., matching or clustering) and the value of an agent i for another agent j is the same as the value of j for i . In contrast, in the graph problems that we consider, the weights are assumed to be *asymmetric*; the weight of an edge is given by the sum of weights of incident vertices.² This makes the results markedly different. Another important distinction is that most of the aforementioned works operate in the setting where the edge weights satisfy the triangle inequality, whereas we impose no such restriction. Caragiannis et al. (2016) study one-side matching settings in metric spaces, and thus their work also falls into this category, but is quite distinct as they focus on cost objectives rather than welfare.

For general social choice settings (i.e., voting), the distortion of ordinal algorithms has been studied in a long list of papers, e.g., see (Procaccia & Rosenschein, 2006; Anshelevich & Postl, 2017; Anshelevich et al., 2018; Boutilier et al., 2015; Caragiannis et al., 2017; Benade et al., 2017; Caragiannis et al., 2018; Fain et al., 2019; Filos-Ratsikas & Miltersen, 2014; Goel et al., 2017; Munagala & Wang, 2019; Feldman et al., 2016; Gkatzelis et al., 2020; Caragiannis et al., 2022). Most of the related work considers the standard case where only ordinal information is given, with a few notable exceptions (Abramowitz et al., 2019; Benade et al., 2017; Bhaskar et al., 2018; Filos-Ratsikas et al., 2020; Filos-Ratsikas & Voudouris, 2021; Anshelevich et al., 2022). In this context, there is also a line of work that considers the effects of limited ordinal information on the distortion, e.g., see (Fain et al., 2019; Kempe, 2020; Gross, Anshelevich, & Xia, 2017). For the one-sided matching problem, a related approach was taken recently by Hosseini et al. (2021), who consider approximations to ordinal objectives via *next-best* queries, i.e., queries which reveal the next best option of an agent, compared to the previous query.

The approach of enhancing the expressiveness of algorithms by equipping them with cardinal queries that we adopt in this paper was first suggested by Amanatidis et al. (2021). It should be noted that this is in nature quite different from another related recent approach proposed by Mandal et al. (2019, 2020), which considers the communication complexity of voting algorithms. In that setting, the algorithm must elicit a limited number of *bits* of

2. Anshelevich et al. (2013) refer to this setting as “Asymmetric Edge-Labeled Graphs” as opposed to “Symmetric Edge-Labeled Graphs”, which is the setting of Anshelevich and Sekar (2016a) and the other works mentioned above.

information from the agents and it is not assumed that the ordinal preferences are already known. Moreover, the agents are allowed within that number of bits to communicate partial information about all of their different values.

Finally, we remark that *outside social choice settings*, the interplay between information and efficiency has been studied in the context of the *stochastic matching* problem (e.g., see (Blum et al., 2020; Chen et al., 2009) and references therein). In that setting, there is a (possibly) weighted stochastic graph, where edges exist with probabilities drawn from some underlying distribution. The existence of a potential edge revealed if it is queried or *probed*, and the goal is to find the probing strategy that results in a maximum-weight matching. Some works compare against the omniscient optimum (Blum et al., 2020; Behnezhad et al., 2019, 2020; Goel & Tripathi, 2012; Costello et al., 2012) while others seek for the best polynomial-time approximation to the optimal probing strategy (Chen et al., 2009; Adamczyk et al., 2015; Bansal et al., 2012; Baveja et al., 2018), with some of these works also focusing their attention to the query-commit variant, where once an edge is probed, it has to be included in the output matching (Chen et al., 2009; Adamczyk et al., 2015; Bansal et al., 2012; Baveja et al., 2018; Goel & Tripathi, 2012; Costello et al., 2012). All of these settings are markedly different from ours, since (a) they assume an underlying probability distribution on the edges of the graph, (b) they do not have access to the ordinal preferences as they do not typically operate in social choice scenarios, (c) they mainly consider randomized algorithms, and (d) similarly to the works mentioned earlier in this section, they mainly use edge costs, effectively symmetric agent valuations.

2. Model Definition

We consider the One-Sided Matching problem, where there is a set of agents N and a set of items A , such that $|N| = |A| = n$. Each agent $i \in N$ has a *valuation function* $v_i : A \rightarrow \mathbb{R}_{\geq 0}$ indicating the agent's value for each item; that is $v_i(j)$ is the value of agent $i \in N$ for item $j \in A$. The valuation functions we consider are either *unrestricted*, in which case the values for the items can be any non-negative real numbers, or *unit-sum*, in which case the sum of values of each agent i for all items is 1: $\sum_{j \in A} v_i(j) = 1$. We denote by $\mathbf{v} = (v_i)_{i \in N}$ the (*cardinal*) *valuation profile* of the agents. Let $Y = (y_i)_{i \in N}$ be a *matching* according to which each agent $i \in N$ is matched to exactly one item $y_i \in A$, such that $y_i \neq y_{i'}$ for every $i \neq i'$. Given a profile \mathbf{v} , the *social welfare* of Y , $\text{SW}(Y|\mathbf{v})$, is the total value of the agents for the items they are matched to according to Y :

$$\text{SW}(Y|\mathbf{v}) = \sum_{i \in N} v_i(y_i).$$

By \mathcal{M} we denote the set of all perfect matchings on our instance. Our goal is to compute a matching $X(\mathbf{v}) = (x_i)_{i \in N}$ with maximum social welfare, i.e.,

$$X(\mathbf{v}) \in \arg \max_{Y \in \mathcal{M}} \text{SW}(Y|\mathbf{v}).$$

In case the valuation functions of the agents are known, then computing $X(\mathbf{v})$ can be done efficiently, e.g., via the Hungarian method (Kuhn, 1956). However, our setting is a bit more restrictive. The exact valuation functions of the agents are their private

information, and they can instead report orderings over the items, which are consistent with their valuations. In particular, every agent i reports a ranking of the items \succ_i such that $a \succ_i b$ if and only if $v_i(a) \geq v_i(b)$ for all $a, b \in A$. Given a valuation profile \mathbf{v} , we denote by $\succ_{\mathbf{v}} = (\succ_i)_{i \in N}$ the *ordinal profile* induced by \mathbf{v} ; observe that different valuation profiles may induce the same ordinal profile. On top of the ordinal preferences of the agents, we can obtain partial access to the valuation profile, by making a number of value queries per agent. In particular, a *value query* takes as input an agent $i \in N$ and an item $j \in A$, and returns the value $v_i(j)$ of agent i for item j . This leads us to the following definition of a deterministic algorithm in our setting.

Definition 1. A *matching algorithm* \mathcal{A}_k takes as input an ordinal profile $(\succ)_{i \in N}$, makes $k \leq n$ value queries per agent and, using $(\succ)_{i \in N}$ as well as the answers to the queries, it computes a matching $\mathcal{A}_k(\succ) \in \mathcal{M}$. If $k = 0$, \mathcal{A} is an *ordinal* algorithm, whereas if $k = n$, \mathcal{A} is a *cardinal* algorithm.

As already mentioned, we can efficiently compute the optimal matching using a cardinal algorithm. However, if an algorithm is allowed to make a limited number $k < n$ of queries per agent, the computed matching might not be optimal. The question then is how well does such an algorithm *approximate* the optimal social welfare of any matching. Approximation here is captured by the notion of *distortion*.

Definition 2. The *distortion* $\text{dist}(\mathcal{A}_k)$ of an algorithm \mathcal{A}_k is the worst-case ratio (over all possible valuation profiles \mathbf{v}) between the social welfare of an optimal matching $X(\mathbf{v})$ and the social welfare of the matching computed by \mathcal{A}_k :

$$\text{dist}(\mathcal{A}_k) = \sup_{\mathbf{v}} \frac{\text{SW}(X(\mathbf{v})|\mathbf{v})}{\text{SW}(\mathcal{A}_k(\succ_{\mathbf{v}})|\mathbf{v})}.$$

Example 1. At this point we introduce an instance with 8 agents and 8 items which will serve as our running example for illustrating how our algorithms work (see Examples 2, 3 and 4). The following table summarizes the underlying valuation profile $\mathbf{v} = (v_i)_{i \in N}$:

| | j_1 | j_2 | j_3 | j_4 | j_5 | j_6 | j_7 | j_8 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| i_1 | 10 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
| i_2 | 12 | 8 | 9 | 5 | 7 | 2 | 2 | 0 |
| i_3 | 18 | 5 | 9 | 2 | 3 | 3 | 4 | 1 |
| i_4 | 8 | 7 | 6 | 5 | 5 | 5 | 5 | 4 |
| i_5 | 24 | 5 | 6 | 1 | 2 | 3 | 4 | 0 |
| i_6 | 20 | 7 | 6 | 3 | 3 | 4 | 2 | 0 |
| i_7 | 9 | 7 | 8 | 4 | 5 | 4 | 5 | 3 |
| i_8 | 15 | 10 | 10 | 2 | 2 | 3 | 2 | 1 |

Of course, the initial input to the algorithm is the ordinal profile $\succ_{\mathbf{v}} = (\succ_i)_{i \in N}$ corresponding to \mathbf{v} . For instance, \succ_1 is determined by the ordered tuple $(j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8)$, whereas \succ_5 is determined by $(j_1, j_3, j_2, j_7, j_6, j_5, j_4, j_8)$. Note that all agents agree on which item is the best, which set contains the second and third best items, and so on. As we will see later (in Example 3), this is an example of a 3-well-structured instance. An optimal matching here is $\{(i_1, j_4), (i_2, j_5), (i_3, j_3), (i_4, j_8), (i_5, j_1), (i_6, j_6), (i_7, j_7), (i_8, j_2)\}$ with social welfare 69. \square

2.1 Warm-Up: Ordinal Algorithms

Before we proceed with our more technical results on tradeoffs between information and distortion, we consider the case of ordinal algorithms. When the valuation functions of the agents are unrestricted, the distortion of any ordinal algorithm is unbounded. To see this, consider any instance that contains two agents who agree on which the most valuable item is. Since only one of them can be matched to this item, it might be the case that the other agent has an arbitrarily large value for it, leading to unbounded distortion. Even for the more restrictive case of unit-sum valuations, however, the distortion of ordinal algorithms can be quite large.

Theorem 1. *The distortion of the best ordinal matching algorithm is $\Theta(n^2)$.*

Proof. For the upper bound, consider any algorithm that outputs a matching so that some agent is matched to her top-ranked item. As the valuations are unit-sum, this agent must have value at least $1/n$ for this item, and thus the social welfare of the matching computed by the algorithm is in turn also at least $1/n$. Since the value of every agent for any item is at most 1, the maximum possible social welfare is upper-bounded by n , and thus the distortion of the algorithm is at most n^2 .

For the lower bound, we assume that n is even; our instance can be easily adjusted for odd n . We consider an instance with set of items $A = \{a, b_1, \dots, b_{n/2}, c_1, \dots, c_{n/2-1}\}$. The ordinal profile is such that, for $i \in \{1, \dots, n/2\}$, agents i and $i + n/2$ have the same ordinal preference \succ_i , defined as

$$a \succ_i b_i \succ_i b_1 \succ_i \dots \succ_i b_{i-1} \succ_i b_{i+1} \succ_i \dots \succ_i b_{n/2} \succ_i c_1 \succ_i \dots \succ_i c_{n/2-1}.$$

Consider any ordinal algorithm which, given as input this profile, outputs a matching $Y = (y_i)_{i \in N}$. We define a valuation profile \mathbf{v} which is consistent with the ordinal profile, and the value of agent $i \in \{1, \dots, n\}$ depends on the structure of Y . For convenience, let s_i denote the second favorite item of agent i , i.e., $s_i = b_i$ if $i \leq n/2$ and $s_i = b_{i-n/2}$ if $i > n/2$.

- If $y_i = a$, then agent i has value $1/n$ for all items;
- If $y_i = s_i$, then agent i has value 1 for a and 0 for every other item;
- Otherwise, agent i has value $1/2$ for a , $1/2$ for s_i and 0 for every other item.

Since only one agent can be matched to a , and everyone else will be matched to an item of value 0, the social welfare of Y computed by the algorithm is $\text{SW}(Y|\mathbf{v}) = 1/n$. However, observe that there exists a matching X with social welfare $\text{SW}(X|\mathbf{v}) \approx n/4$. In particular, we can go through the agents and match each agent $i \in \{1, \dots, n\}$ to s_i if she is not already matched to s_i or to a in Y . This way we will end up with a matching where at least $n/2 - 1$ agents will have value $1/2$ each for the corresponding items. Our claim about the social welfare follows. Therefore, the distortion of any ordinal algorithm is $\Omega(n^2)$. \square

3. Distortion Guarantees for Unconstrained Valuations

Here we present λ -THRESHOLDSTEPFUNCTION (λ -TSF), an algorithm that works for any valuation functions. At a high level, for each agent, we do the following. We first query

Algorithm 1: λ -THRESHOLDSTEPFUNCTION (λ -TSF)

Let $\alpha_\ell := n^{-\ell/(\lambda+1)}$ for $\ell \in \{0, \dots, \lambda\}$.
for every agent $i \in N$ **do**
 Query the value v_i^* of i for her top-ranked item j_i^*
 Let $Q_{i,0} := \{j_i^*\}$
 Set $\tilde{v}_i(j_i^*) := \alpha_0 \cdot v_i^* = v_i^*$
 for every $\ell \in \{1, \dots, \lambda\}$ **do**
 Using binary search, compute $Q_{i,\ell} := \{j \in A : v_i(j) \in [\alpha_\ell \cdot v_i^*, \alpha_{\ell-1} \cdot v_i^*]\}$
 Set $\tilde{v}_i(j) := \alpha_\ell \cdot v_i^*$ for every $j \in Q_{i,\ell}$
 Let $Q_i := \bigcup_{\ell=0}^{\lambda} Q_{i,\ell}$
 Set $\tilde{v}_i(j) := 0$ for every $j \in A \setminus Q_i$
return a matching $Y \in \arg \max_{Z \in \mathcal{M}} \text{SW}(Z | \tilde{\mathbf{v}})$.

the agent’s value for her highest ranked item. Then, we partition the items into $\lambda + 1$ sets, so that the agent’s value for all the items in a set is lower-bounded by a carefully selected quantity related to the agent’s top value. Based on this partition, we then define a new *simulated valuation function* for the agent, where the value of an item is equal to the lower bound that corresponds to the set the item belongs to. Finally, we compute a maximum weight matching with respect to the simulated valuation functions. See Algorithm 1.

For each i and ℓ , in order to find i ’s least preferred item that she values at least $\alpha_\ell \cdot v_i^*$, we run a standard binary search on \succ_i . It is known that each such binary search requires $1 + \log_2 n$ queries.

Example 2. Suppose that we run 2-TSF on the instance introduced in Example 1. Since $n = 8$ here, we have $\alpha_0 = 1$, $\alpha_1 = 1/2$, and $\alpha_2 = 1/4$. For each agent i , the algorithm constructs the simulated valuation function \tilde{v}_i . We will go through this in detail for agent i_1 . The algorithm first asks agent i_1 about her favorite item j_1 , and sets $\tilde{v}_1(j_1) = 10$. Then, the algorithm finds the last item (with respect to \succ_1) for which i_1 has value at least $\tilde{v}_1(j_1)/\alpha_1 = 5$ using binary search on the array $(j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8)$. The exact sequence of queries depends on the specific implementation of binary search, e.g., assuming that binary search considers the left median in arrays of even length, the algorithm will ask about j_4 , j_6 , and j_5 , in that order, and decide that j_5 is the item in question. The simulated value of every item after j_1 up to and including j_5 is then set to $\tilde{v}_1(j_1)/\alpha_1 = 5$. Next, the algorithm finds the last item after j_5 (with respect to \succ_1) for which i_1 has value at least $\tilde{v}_1(j_1)/\alpha_2 = 2.5$ using binary search on (j_6, j_7, j_8) . Assuming the same binary search variant as before, the algorithm will ask about j_7 and j_8 , and decide that j_7 is the item in question. The simulated value of every item after j_5 up to and including j_7 is then set to $\tilde{v}_1(j_1)/\alpha_2 = 2.5$. For all remaining items (here just for j_8), the simulated value is set to 0. Note that for instances this small, we end up asking about most items and that we do not use many of these answers for defining \tilde{v}_1 . Although this seems wasteful, as n grows, the fraction of the values the algorithm learns becomes very small and the answers it ignores would make no difference for our analysis asymptotically. The table summarizes the simulated valuation profile $\tilde{\mathbf{v}} = (\tilde{v}_i)_{i \in N}$:

| | j_1 | j_2 | j_3 | j_4 | j_5 | j_6 | j_7 | j_8 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| i_1 | 10 | 5 | 5 | 5 | 5 | 2.5 | 2.5 | 0 |
| i_2 | 12 | 6 | 6 | 5 | 6 | 0 | 0 | 0 |
| i_3 | 18 | 4.5 | 9 | 0 | 0 | 0 | 0 | 0 |
| i_4 | 8 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| i_5 | 24 | 0 | 6 | 0 | 0 | 0 | 0 | 0 |
| i_6 | 20 | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| i_7 | 9 | 4.5 | 4.5 | 2.25 | 4.5 | 2.25 | 4.5 | 2.25 |
| i_8 | 15 | 7.5 | 7.5 | 0 | 0 | 0 | 0 | 0 |

An optimal matching with respect to $\tilde{\mathbf{v}}$, is $\{(i_1, j_4), (i_2, j_5), (i_3, j_3), (i_4, j_6), (i_5, j_1), (i_6, j_8), (i_7, j_7), (i_8, j_2)\}$ with simulated social welfare 60. The actual social welfare of this matching with respect to \mathbf{v} is 66; thus, it is suboptimal but still approximates the maximum social welfare very well. \square

The next theorem follows (asymptotically) from the more general Theorem 9, which is stated in Section 6 and applies to a number of well-known graph problems. To aid the reader, we also include a self-contained, cleaner proof of Theorem 2 which applies only to One-Sided Matching and gives a slightly better bound.

Theorem 2. λ -TSF makes $1 + \lambda + \lambda \log n$ queries per agent and achieves a distortion of $2n^{1/(\lambda+1)}$.

Proof. Consider an arbitrary valuation profile \mathbf{v} . Let $X = X(\mathbf{v})$ be an optimal matching, and Y be the solution returned by λ -TSF; recall that Y maximizes the social welfare according to the simulated valuation functions \tilde{v}_i . Let S be the set of agents i such that $x_i \in Q_i$, and $\bar{S} = N \setminus S$. Then, the optimal social welfare can be written as

$$\text{SW}(X|\mathbf{v}) = \sum_{i \in N} v_i(x_i) = \sum_{i \in \bar{S}} v_i(x_i) + \sum_{i \in S} v_i(x_i).$$

We will bound the two terms separately. We begin with the first one:

$$\sum_{i \in \bar{S}} v_i(x_i) < \alpha_\lambda \sum_{i \in \bar{S}} v_i^* \leq \alpha_\lambda \cdot n \cdot \max_{i \in N} v_i^* \leq \alpha_\lambda \cdot n \cdot \text{SW}(Y|\tilde{\mathbf{v}}) \leq \alpha_\lambda \cdot n \cdot \text{SW}(Y|\mathbf{v}).$$

The first inequality follows directly by the definition of Q_i . The second inequality follows since $\bar{S} \subseteq N$. The third inequality follows since Y maximizes the social welfare according to the profile \mathbf{v} and the algorithm has queried all agents for their most-preferred items. Finally, the last inequality follows since the simulated values of an agents are lower bounds on her true ones.

For the second term, let S_ℓ be the restriction of S on agents for whom $x_i \in Q_{i,\ell}$, $\ell \in \{0, \dots, \lambda\}$. Then,

$$\sum_{i \in S} v_i(x_i) = \sum_{\ell=0}^{\lambda} \sum_{i \in S_\ell} v_i(x_i).$$

Now, let us assume that $\lambda > 0$; we will deal with the simpler case where $\lambda = 0$ later. By definition, for any $\ell \in \{1, \dots, \lambda\}$ and any $j \in Q_{i,\ell}$, we have that $v_i(j) \leq \alpha_{\ell-1} \cdot v_i^* = \frac{\alpha_{\ell-1}}{\alpha_\ell} \cdot \alpha_\ell \cdot v_i^* = \tilde{v}_i(j)/\alpha_1$. Also, for $Q_{i,0} = \{j_i^*\}$, we have $v_i(j_i^*) = \tilde{v}_i(j_i^*) \leq \tilde{v}_i(j_i^*)/\alpha_1$. Hence,

$$\sum_{i \in S} v_i(x_i) \leq \alpha_1^{-1} \sum_{\ell=0}^{\lambda} \sum_{i \in S_\ell} \tilde{v}_i(x_i) \leq \alpha_1^{-1} \sum_{i \in N} \tilde{v}_i(x_i) \leq \alpha_1^{-1} \text{SW}(Y|\tilde{\mathbf{v}}) \leq \alpha_1^{-1} \text{SW}(Y|\mathbf{v}).$$

The second inequality follows by considering all agents. The third inequality follows from the optimality of Y with respect to the simulated valuation functions. Finally, the last inequality follows since the simulated values of an agents are lower bounds on her true ones.

Now we can put everything together:

$$\text{SW}(X|\mathbf{v}) \leq (\alpha_\lambda \cdot n + \alpha_1^{-1}) \text{SW}(Y|\mathbf{v}) = 2n^{1/(\lambda+1)} \cdot \text{SW}(Y|\mathbf{v}), \quad (1)$$

and this settles the bound on the distortion when $\lambda > 0$.

When $\lambda = 0$, we clearly have that

$$\sum_{i \in S} v_i(x_i) \leq \sum_{i \in S} \tilde{v}_i(x_i) \leq \text{SW}(Y|\mathbf{v}).$$

Then, the analog of (1) is

$$\text{SW}(X|\mathbf{v}) \leq (\alpha_\lambda \cdot n + 1) \text{SW}(Y|\mathbf{v}) = 2n^{1/(\lambda+1)} \cdot \text{SW}(Y|\mathbf{v}).$$

This concludes the proof. \square

By appropriately setting the value of λ , we obtain several tradeoffs between the distortion and the number of queries per agent. In particular, we have the following statement.

Corollary 1. *We can achieve*

- *distortion $O(n)$ by making one query per agent;*
- *distortion $O(n^{1/k})$ for any constant integer k by making $O(\log n)$ queries per agent;*
- *distortion $O(1)$ by making $O(\log^2 n)$ queries per agent.*

3.1 Well-Structured Instances

We now consider instances in which the agents exhibit quite similar ordinal preferences. For any positive integer k , we define the class of k -well-structured (k -WS) instances. Let $\varepsilon \in (0, 1]$ be a constant (e.g., $\varepsilon = 1/2$). In a k -WS instance, the set of items can be partitioned into $k + 1$ sets A_1, \dots, A_k, A_{k+1} such that

$$|A_1| = 1 \quad \text{and} \quad |A_\ell| = \left\lceil \varepsilon \cdot n^{(\ell-1)/k} \right\rceil \quad \text{for all } \ell \in \{2, \dots, k\},$$

and every agent i has the ordinal preference

$$\langle A_1 \rangle_i \succ_i \langle A_2 \rangle_i \succ_i \dots \succ_i \langle A_k \rangle_i \succ_i \langle A_{k+1} \rangle_i,$$

Algorithm 2: k -FIXEDMAXMATCHING (k -FMM)

for every $\ell \in \{1, \dots, k\}$ **do**
 Query the value $u_i(\ell)$ of each agent i for her least-preferred item in A_ℓ
for every agent $i \in N$ **do**
 Define the simulated valuation function \tilde{v}_i :
 • $\tilde{v}_i(j) := u_i(\ell)$ for every $j \in A_\ell, \ell \in \{1, \dots, k\}$
 • $\tilde{v}_i(j) := 0$ for every $j \in A_{k+1}$
return a matching $Y \in \arg \max_Z \text{SW}(Z|\tilde{\mathbf{v}})$

where $\langle A_\ell \rangle_i$ denotes some ordering of the items in set A_ℓ which depends on agent i ; that is, different agents may order the items in A_ℓ differently. Observe that an instance in which all agents have the same ranking over the items is a k -WS instance for every k . We will use such instances in our lower bounds in the next section. For simplicity, since all our statements are asymptotic, we drop the ceiling notation as is we only dealt with instances where $\varepsilon \cdot n^{(\ell-1)/k}$ is integral for all ℓ .

For the class of k -WS instances, we present a quite simple algorithm, which we call k -FIXEDMAXMATCHING (k -FMM); see Algorithm 2. This algorithm achieves a distortion of $O(k \cdot n^{1/k})$ by making k queries per agent.

Example 3. First, observe that the instance introduced in Example 1 is a 3-well-structured instance with $\varepsilon = 1$, $A_1 = \{j_1\}$, $A_2 = \{j_2, j_3\}$, $A_3 = \{j_4, j_5, j_6, j_7\}$, and $A_4 = \{j_8\}$ (verifying the set sizes is just a matter of simple calculations). That is, everyone agrees that j_1 is the best item, that j_2 and j_3 are the second and third best items, but not necessarily in that order, and so on.

Suppose now that we run 3-FMM on this instance. For each agent i , the algorithm constructs the simulated valuation function \tilde{v}_i by querying about i 's least favorite item (with respect to \succ_i) in each of A_1 , A_2 , and A_3 . The answers to these queries then become the simulated values for all the items within each set, while all the items in A_4 have simulated value 0. For instance, the algorithm queries i_1 about j_1 , j_3 , and j_7 and the simulated values are $\tilde{v}_1(j_1) = 10$, $\tilde{v}_1(j_2) = \tilde{v}_1(j_3) = 7$, $\tilde{v}_1(j_4) = \tilde{v}_1(j_5) = \tilde{v}_1(j_6) = \tilde{v}_1(j_7) = 3$, and $\tilde{v}_1(j_8) = 0$. The following table summarizes the simulated valuation profile $\tilde{\mathbf{v}} = (\tilde{v}_i)_{i \in N}$:

| | j_1 | j_2 | j_3 | j_4 | j_5 | j_6 | j_7 | j_8 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| i_1 | 10 | 7 | 7 | 3 | 3 | 3 | 3 | 0 |
| i_2 | 12 | 8 | 8 | 2 | 2 | 2 | 2 | 0 |
| i_3 | 18 | 5 | 5 | 2 | 2 | 2 | 2 | 0 |
| i_4 | 8 | 6 | 6 | 5 | 5 | 5 | 5 | 0 |
| i_5 | 24 | 5 | 5 | 1 | 1 | 1 | 1 | 0 |
| i_6 | 20 | 6 | 6 | 2 | 2 | 2 | 2 | 0 |
| i_7 | 9 | 7 | 7 | 4 | 4 | 4 | 4 | 0 |
| i_8 | 15 | 10 | 10 | 2 | 2 | 2 | 2 | 0 |

An optimal matching with respect to $\tilde{\mathbf{v}}$, is $\{(i_1, j_4), (i_2, j_2), (i_3, j_6), (i_4, j_7), (i_5, j_1), (i_6, j_8), (i_7, j_5), (i_8, j_3)\}$ with simulated social welfare 56. The social welfare of this matching with respect to the true valuation profile \mathbf{v} is 61. \square

Theorem 3. *For the class of k -well-structured instances with $k \geq 1$, k -FMM makes k queries per agent and achieves a distortion of $O(k \cdot n^{1/k})$.*

Proof. Consider an arbitrary k -WS instance with valuation profile \mathbf{v} . For $\ell \in \{1, \dots, k\}$, denote by S_ℓ the set of $|A_\ell| = n^{(\ell-1)/k}$ agents with the highest values for the last item in A_ℓ (breaking ties arbitrarily). Since there exists a matching of the items in A_ℓ to the agents of S_ℓ , the algorithm maximizes the simulated welfare, and $v_i(j) \geq \tilde{v}_i(j)$ for every agent i and item j , the social welfare of the matching Y computed by the algorithm is

$$\begin{aligned} \text{SW}(Y|\mathbf{v}) &\geq \sum_{i \in S_\ell} u_i(\ell) \\ &\geq n^{(\ell-1)/k} \cdot \min_{i \in S_\ell} u_i(\ell), \end{aligned}$$

for every $\ell \in \{1, \dots, k\}$. Observe that since A_1 consists of just one item, the inequality for $\ell = 1$ can be simplified to

$$\text{SW}(Y|\mathbf{v}) \geq \max_i u_i(1).$$

Now, let X be an optimal matching, and denote by x_j the agent matched to item $j \in A$. Then,

$$\begin{aligned} \text{SW}(X|\mathbf{v}) &= \sum_{\ell=1}^{k+1} \sum_{j \in A_\ell} v_{x_j}(j) = \sum_{j \in A_1} v_{x_j}(j) + \sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} v_{x_j}(j) \\ &\leq \max_i u_i(1) + \sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} u_{x_j}(\ell - 1), \end{aligned}$$

where the inequality follows since A_1 consists of just one item, and the values of the agents for every item in A_ℓ are at most their values for the last item in $A_{\ell-1}$ (since all items in $A_{\ell-1}$ are ranked higher than the items in A_ℓ). Let us focus on the right-most term of the above expression. We have

$$\sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} u_{x_j}(\ell - 1) = \sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} \sum_{x_j \in S_{\ell-1}} u_{x_j}(\ell - 1) + \sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} \sum_{x_j \notin S_{\ell-1}} u_{x_j}(\ell - 1).$$

By the definition of $S_{\ell-1}$, for every agent $i \notin S_{\ell-1}$, it holds that $u_{x_j}(\ell - 1) \leq \min_{i \in S_{\ell-1}} u_i(\ell - 1)$, and thus

$$\sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} u_{x_j}(\ell - 1) \leq \sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} \sum_{x_j \in S_{\ell-1}} u_{x_j}(\ell - 1) + \sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} \sum_{x_j \notin S_{\ell-1}} \min_{i \in S_{\ell-1}} u_i(\ell - 1).$$

Observe that the internal double sum of the first term can sum over at most all agents in $S_{\ell-1}$, while the internal double sum of the second term can sum over at most $|A_\ell|$ agents.

Using the fact that $|A_{k+1}| \leq n$ and the lower bounds for the social welfare of the matching computed by the algorithm, we obtain

$$\begin{aligned}
\sum_{\ell=2}^{k+1} \sum_{j \in A_\ell} u_{x_j}(\ell-1) &\leq \sum_{\ell=2}^{k+1} \sum_{i \in S_{\ell-1}} u_i(\ell-1) + \sum_{\ell=2}^{k+1} |A_\ell| \min_{i \in S_{\ell-1}} u_i(\ell-1) \\
&= \sum_{\ell=2}^{k+1} \sum_{i \in S_{\ell-1}} u_i(\ell-1) + \sum_{\ell=2}^k n^{(\ell-1)/k} \cdot \min_{i \in S_{\ell-1}} u_i(\ell-1) + |A_{k+1}| \min_{i \in S_k} u_i(k) \\
&\leq \sum_{\ell=1}^k \sum_{i \in S_\ell} u_i(\ell) + n^{1/k} \sum_{\ell=1}^{k-1} n^{(\ell-1)/k} \cdot \min_{i \in S_\ell} u_i(\ell) + n \cdot \min_{i \in S_k} u_i(k) \\
&\leq (k + k \cdot n^{1/k}) \cdot \text{SW}(Y|\mathbf{v}).
\end{aligned}$$

Putting everything together, we obtain the theorem. \square

We conclude our discussion on k -WS instances with a lower bound $\Omega(n^{1/k})$ on the distortion of the $(k-1)$ -TSF algorithm; here, the value of the parameter λ is chosen to be $k-1$ because of the structure of k -WS instances. Combined together with the $O(k \log n)$ queries that it requires to operate, we have that when k is sub-logarithmic, the k -FMM algorithm presented above matches the distortion of $(k-1)$ -TSF on k -WS instances, using a factor of $\log n$ less queries per agent.

Theorem 4. *For every constant $k \geq 1$, there exists a k -well-structured instance such that the distortion of $(k-1)$ -TSF when given this instance as input is $\Omega(n^{1/k})$.*

Proof. Consider an instance in which all agents have the same ordinal preference over the items, which can be easily seen to be a k -WS instance for every $k \geq 1$. Let us now define the cardinal values which are revealed when the $(k-1)$ -TSF algorithm queries the agents:

- $\alpha_{\ell-1} = n^{-(\ell-1)/k}$ for queries about items in A_ℓ , $\ell \leq k$;
- 0 for queries about items in A_{k+1} .

Because the rankings of the agents for the items and the revealed information due to the queries of the algorithm are the same among the agents, the algorithm will define the same simulated valuation function for all agents. In particular, based on the revealed values, we have that $Q_{i,\ell-1} = A_\ell$ for every $\ell \in \{1, \dots, k\}$. By considering a valuation profile \mathbf{v} according to which an agent that is matched to an item in set A_ℓ has value $\alpha_{\ell-1}$ for it, we have that the social welfare of the matching Y computed by the algorithm is

$$\text{SW}(Y|\mathbf{v}) = \sum_{\ell=1}^k |A_\ell| \cdot \alpha_{\ell-1} = 1 + \varepsilon \sum_{\ell=2}^k n^{(\ell-1)/k} \cdot n^{-(\ell-1)/k} = 1 + \varepsilon \cdot (k-1) \leq k.$$

Now, observe that when binary search is restricted to run over only the items in the set A_ℓ , $\ell \in \{2, \dots, k+1\}$, it does not query all the items in A_ℓ ; in particular, because of the way binary search operates, the value of the first $|A_\ell|/2$ items therein will never be revealed. Hence, even if the algorithm matches the agents to items for which they have not been

queried for, there must exist a matching X such that for every $\ell \in \{2, \dots, k+1\}$ the first $|A_\ell|/2$ items of A_ℓ are matched to agents different than the ones chosen by the algorithm, who have not been queried for their values. By setting the real values of the agents for these items to be $\alpha_{\ell-2}$, and observing that $|A_{k+1}| = \xi n$ for some constant $\xi \in (0, 1)$, we have

$$\begin{aligned} \text{SW}(X|\mathbf{v}) &\geq \sum_{\ell=2}^{k+1} \frac{|A_\ell|}{2} \cdot \alpha_{\ell-2} \\ &= \frac{\varepsilon}{2} \sum_{\ell=2}^k n^{(\ell-1)/k} \cdot n^{-(\ell-2)/k} + \frac{\xi}{2} \cdot n \cdot n^{-(k-1)/k} \\ &\geq \frac{\min\{\varepsilon, \xi\}}{2} \cdot k \cdot n^{1/k}. \end{aligned}$$

Hence, the distortion is $\Omega(n^{1/k})$. □

4. Lower Bounds

In this section we show unconditional lower bounds for algorithms for One-Sided Matching which are allowed to make at most $k \geq 1$ queries per agent. We present a generic matching instance which can be fine-tuned to yield lower bounds for both unrestricted and unit-sum valuation functions. Let \mathcal{V} denote any of these two classes of valuation functions.

Let $\delta_{\mathcal{V}}(k) \leq 1/k$ be a function of k , and $\varepsilon \in (0, 1/2)$ be some constant. We want to define an instance in which the n items are partitioned into $k+2$ sets $A_1, \dots, A_{k+1}, B = A \setminus (\bigcup_{\ell \in [k+1]} A_\ell)$ such that

$$|A_1| = 1 \text{ and } |A_\ell| = \varepsilon \cdot n^{(\ell-1)\delta_{\mathcal{V}}(k)} \text{ for every } \ell \in \{2, \dots, k+1\}.$$

Note that because we have restricted the possible values of $\delta_{\mathcal{V}}(k)$ to be at most $1/k$ and have chosen $\varepsilon < 1/2$, these sets of items can be defined; in particular, $|A_{k+1}| \leq \varepsilon n$. We assume that n is large enough so that $n > 2 \sum_{\ell=1}^{k+1} |A_\ell|$ and that is such that the cardinalities are indeed integers; the latter is only assumed to simplify the notation. We use $\langle T \rangle$ to denote some fixed arbitrary ranking of the elements of set T (which is common for all agents). Given that, we define the ordinal preference of every agent $i \in N$ to be

$$\langle A_1 \rangle \succ_i \langle A_2 \rangle \succ_i \dots \succ_i \langle A_k \rangle \succ_i \langle A_{k+1} \rangle \succ_i \langle B \rangle.$$

We reveal the following information, depending on the queries of the algorithm:

- For every $\ell \in \{1, \dots, k+1\}$, any query for some item in A_ℓ reveals a value of $|A_\ell|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)}$;
- Every query for some item in B reveals a value of 0.

Observe that for $\delta_{\mathcal{V}}(k) = 1/k$, we have actually defined a k -well-structured instance in which all agents have the same ordinal preference. As we will see below, this choice of $\delta_{\mathcal{V}}$ in fact yields the best lower bound if \mathcal{V} is the class of unrestricted valuations. However, when \mathcal{V} is the class of unit-sum valuations, we will have to choose $\delta_{\mathcal{V}}$ differently.

Next, we define two types of conditional valuation functions that an agent i may have, depending on the behavior of the algorithm. These functions have to be consistent to the information that is revealed by the queries of the algorithm. Let $\xi \in (0, 1]$ be some constant.

(T1) If there exists $r \in \{1, \dots, k+1\}$, such that i is *not* queried for any item in A_r and she *does not* get an item from A_r either, then i 's values are

- at least $\xi \cdot |A_{r-1}|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)}$ for each item in A_r if $r \geq 2$;
- at least ξ for the item in A_1 if $r = 1$;
- $|A_\ell|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)}$ for every item in A_ℓ , for $\ell \in \{1, \dots, k+1\} \setminus \{r\}$;
- 0 for every item in B .

(T2) If i is queried for some item in k different sets out of A_1, \dots, A_{k+1} , then her values are

- $|A_\ell|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)}$ for every item in A_ℓ , for $\ell \in \{1, \dots, k+1\}$;
- at most $|A_{k+1}|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)}$ for every item in B .

Observe that the conditions specified in (T1) and (T2) capture all possible cases about the queries of the algorithm and the possible assignments of the items to the agents.

Theorem 5. *Let \mathcal{V} be the class of unrestricted or unit-sum valuation functions. If there exists a function $\delta_{\mathcal{V}}(k) \leq 1/k$ such that it is possible to define valuation functions in \mathcal{V} of types (T1) and (T2), the distortion of any matching algorithm which makes k queries per agent is $\Omega(\frac{1}{k} \cdot n^{\delta_{\mathcal{V}}(k)})$.*

Proof. Observe that if the values of the agents for all items in a set A_ℓ are consistent to the revealed values, then the total value for all items in set A_ℓ is equal to $|A_\ell| \cdot |A_\ell|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)} = n^{-\delta_{\mathcal{V}}(k)}$. Since there are valuation functions of types (T1) and (T2), we can indeed define a valuation profile \mathbf{v} so that the value of every agent for the item she is matched to by the algorithm is exactly the value she would reveal if she was queried for it. So, the items of every set A_ℓ , $\ell \in \{1, \dots, k+1\}$, contribute exactly $n^{-\delta_{\mathcal{V}}(k)}$ to the social welfare of the matching Y computed by the algorithm, while the items in B contribute 0. Hence,

$$\text{SW}(Y|\mathbf{v}) = (k+1) \cdot n^{-\delta_{\mathcal{V}}(k)}.$$

Hence, to show the desired bound on the distortion of the algorithm, it suffices to show that there is always a matching with social welfare $\Omega(1)$.

This is clearly the case when there exists an agent who is not queried for the item in A_1 and is not given this item, since her value for it in such a case can be set to be at least ξ using a function of type (T1) for $r = 1$. Therefore, since this item can only be given to one agent, in the following we assume that there is *at most one* agent who is not queried for it, and if such an agent exists, she must be given the item.

Consider the set S_2 of agents who are not queried for any item in A_2 and also do not get any item in A_2 . If $|S_2| \geq |A_2|$, then by defining the valuation function of every agent in S_2

to be of type (T1) for $r = 2$, we can obtain a matching (in which exactly $|A_2|$ agents in S_2 are given a different item from A_2) with social welfare at least $|A_2| \cdot \xi \cdot |A_1|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)} = \xi$. The latter follows by the fact that $|A_t| \cdot |A_{t-1}|^{-1} = n^{\delta_{\mathcal{V}}(k)}$ for all $t \geq 2$. Consequently, it must be $|S_2| < |A_2|$. This further implies that there are at least $n - 1 - 2|A_2|$ agents who are queried for some item in each of A_1 and A_2 , and do not get any item in $A_1 \cup A_2$. Let L_2 be the set of these agents.

Consider the set $S_3 \subseteq L_2$ of agents who are not queried for any item in A_3 and also do not get any item in A_3 . Like in the case of A_3 and S_3 above, if $|S_3| \geq |A_3|$, then by defining the valuation function of every agent in S_3 to be of type (T1) for $r = 3$, we can obtain a matching (in which $|A_3|$ of the agents in S_3 are given a different item from A_3) with social welfare at least $|A_3| \cdot \xi \cdot |A_2|^{-1} \cdot n^{-\delta_{\mathcal{V}}(k)} = \xi$. So, it must be $|S_3| < |A_3|$, which implies that there are $|L_2| - |A_3| - |S_3| \geq |L_2| - 2|A_3|$ agents in L_2 who are queried for some item in each of A_1, A_2 and A_3 , and do not get any item in $A_1 \cup A_2 \cup A_3$. Let L_3 be the set of these agents.

By induction, this process leads to the existence of the set L_k of agents who have been queried for some item in each of A_1, \dots, A_k , such that $|L_k| \geq n - 1 - 2 \sum_{\ell=1}^k |A_\ell|$. Since

$$n > 2 \sum_{\ell=1}^{k+1} |A_\ell| \Rightarrow n - 1 - 2 \sum_{\ell=1}^k |A_\ell| - |A_{k+1}| \geq |A_{k+1}|,$$

there are $|A_{k+1}|$ agents who have not been queried for any item in A_{k+1} and do not get any item in A_{k+1} . Thus, by setting their valuation functions to be of type (T1) for $r = k + 1$, we can construct a matching with welfare at least ξ , completing the proof. \square

Theorem 5 is actually quite powerful and allows us to prove lower bounds for both unrestricted and unit-sum valuation functions. In particular, it reduces the problem to finding the largest possible $\delta_{\mathcal{V}}(k) \leq 1/k$, such that valuation functions in \mathcal{V} of types (T1) and (T2) can be defined.

Theorem 6. *For unconstrained valuation functions, the distortion of any matching algorithm which makes k queries per agent is $\Omega(\frac{1}{k} \cdot n^{1/k})$.*

Proof. It is straightforward to observe that it is indeed possible to define unconstrained valuation functions of types (T1) and (T2) for the function $\delta_{\infty}(k) = 1/k$. In particular, the valuation functions are such that

$$u_1(j) = \begin{cases} n^{-(r-1)/k}, & \text{if } j \in A_r \\ n^{-\ell/k}, & \text{if } j \in A_\ell, \ell \neq r \\ 0, & \text{if } j \in B \end{cases}$$

and

$$u_2(j) = \begin{cases} n^{-\ell/k}, & \text{if } j \in A_\ell \\ 0, & \text{if } j \in B \end{cases}$$

where ℓ is a generic index, while r in the definition of u_1 is an index that corresponds to a set A_r such that i is not queried for any item in it and she does not get an item from it either.

Hence, by Theorem 5, any matching algorithm has distortion $\Omega(\frac{1}{k} \cdot n^{1/k})$. \square

For unit-sum valuations, we have the following bound.

Theorem 7. *Let $\xi \in (0, 1)$ be a constant. For unit-sum valuation functions, the distortion of any matching algorithm which makes $k \leq (1 - \xi)n^{1/(k+1)}$ queries per agent is $\Omega(\frac{1}{k} \cdot n^{1/(k+1)})$.*

Proof. Let $\varepsilon \in (0, 1/2)$ be the constant used for defining the sets A_1, \dots, A_{k+1} . We will show that for $\delta_1(k) = 1/(k+1)$, the following two valuation functions satisfy the unit-sum normalization and are of types (T1) and (T2), respectively. Then, the statement will follow by Theorem 5 by substituting $\delta_1(k)$. The functions are defined as

$$u_1(j) = \begin{cases} \frac{1-k \cdot n^{-1/(k+1)}}{\varepsilon n^{(r-1)/(k+1)}}, & \text{if } j \in A_r \\ \varepsilon^{-1} n^{-\ell/(k+1)}, & \text{if } j \in A_\ell, \ell \neq r \\ 0, & \text{if } j \in B \end{cases}$$

and

$$u_2(j) = \begin{cases} \varepsilon^{-1} n^{-\ell/(k+1)}, & \text{if } j \in A_\ell \\ \frac{1-(k+1) \cdot n^{-1/(k+1)}}{|B|}, & \text{if } j \in B \end{cases}$$

where ℓ is a generic index, while r in the definition of u_1 is an index that corresponds to a set A_r such that i is not queried for any item in it and she does not get an item from it either.

First, let us verify that both functions satisfy the unit-sum assumption. By the definition of $\delta_1(k)$, we have that $|A_\ell| = \varepsilon n^{(\ell-1)/(k+1)}$ for all $\ell \in \{1, \dots, k+1\}$. Therefore,

$$\begin{aligned} \sum_{j \in A} u_1(j) &= |A_r| \cdot \frac{1-k \cdot n^{-1/(k+1)}}{\varepsilon n^{(r-1)/(k+1)}} + \sum_{\ell=1}^{k+1} |A_\ell| \cdot \varepsilon^{-1} n^{-\ell/(k+1)} - |A_r| \cdot \varepsilon^{-1} n^{-r/(k+1)} \\ &= 1 - k \cdot n^{-1/(k+1)} + k \cdot n^{-1/(k+1)} = 1, \end{aligned}$$

and

$$\begin{aligned} \sum_{j \in A} u_2(j) &= \sum_{\ell=1}^{k+1} |A_\ell| \cdot \varepsilon^{-1} n^{-\ell/(k+1)} + |B| \cdot \frac{1-(k+1) \cdot n^{-1/(k+1)}}{|B|} \\ &= (k+1) \cdot n^{-1/(k+1)} + 1 - (k+1) \cdot n^{-1/(k+1)} = 1. \end{aligned}$$

Next, we will show that u_1 is of type (T1). It suffices to show that the values satisfy the corresponding conditions. We have

- For every item $j \in A_r$, if $r \geq 2$:

$$\begin{aligned} u_1(j) &= \frac{1-k \cdot n^{-1/(k+1)}}{\varepsilon n^{(r-1)/(k+1)}} = \varepsilon^{-1} (n^{-(r-1)/(k+1)} - k \cdot n^{-r/(k+1)}) \\ &\geq \varepsilon^{-1} \cdot \xi \cdot n^{-(r-1)/(k+1)} = \varepsilon^{-1} \cdot \xi \cdot n^{-(r-2)/(k+1)} \cdot n^{-1/(k+1)} \\ &= \xi \cdot |A_{r-1}|^{-1} \cdot n^{-\delta_1(k)}. \end{aligned}$$

- For the item $j \in A_1$: $u_1(j) = \varepsilon^{-1}(1 - k \cdot n^{-1/(k+1)}) \geq \varepsilon^{-1}\xi \geq \xi$.
- For every item $j \in A_\ell$, $\ell \neq r$:

$$u_1(j) = \varepsilon^{-1}n^{-\ell/(k+1)} = \varepsilon^{-1}n^{-(\ell-1)/(k+1)} \cdot n^{-1/(k+1)} = |A_\ell|^{-1} \cdot n^{-\delta_1(k)}.$$

- For every item $j \in B$: $u_1(j) = 0$.

Finally, we will show that u_2 is of type (T2). Similarly to the above case, we have

- For every item $j \in A_\ell$:

$$u_2(j) = \varepsilon^{-1}n^{-\ell/(k+1)} = \varepsilon^{-1}n^{-(\ell-1)/k} \cdot n^{-1/(k+1)} = |A_\ell|^{-1} \cdot n^{-\delta_1(k)}.$$

- For every item $j \in B$:

$$\begin{aligned} u_2(j) &= \frac{1 - (k+1) \cdot n^{-1/(k+1)}}{|B|} = \frac{1 - (k+1) \cdot n^{-1/(k+1)}}{n - \sum_{\ell=1}^{k+1} n^{(\ell-1)/(k+1)}} \\ &\leq \frac{1 - (k+1) \cdot n^{-1/(k+1)}}{n - (k+1)n^{k/(k+1)}} = n^{-1} = \varepsilon \cdot \varepsilon^{-1} \cdot n^{-k/(k+1)} \cdot n^{-1/(k+1)} \\ &= \varepsilon \cdot |A_{k+1}|^{-1} \cdot n^{-\delta_1(k)} \leq |A_{k+1}|^{-1} \cdot n^{-\delta_1(k)}, \end{aligned}$$

where the first inequality follows by the fact that $n^{x/(k+1)}$ is increasing in x . \square

By appropriately setting the value of k in Theorems 6 and 7, we establish that it is impossible to achieve constant distortion without an almost logarithmic number of queries.

Corollary 2. *Any matching algorithm allowed to make at most*

- *a constant number k queries per agent has distortion $\Omega(n^{1/k})$ when the valuation functions are unrestricted, and $\Omega(n^{1/(k+1)})$ when they are unit-sum;*
- *$o\left(\frac{\log n}{\log \log n}\right)$ queries per agent has distortion $\omega(\log \log n)$.*

5. Two Queries for Unit-Sum Valuations

In this section, we present the FIRSTPOSITIONADAPTIVE algorithm (FPA), which makes at most two queries per agent and achieves a distortion of $O(n^{2/3}\sqrt{\log n})$, when the valuation functions are unit-sum. First, we query each agent for their most-preferred item. Then, depending on whether the *maximum revealed value* by these queries is at least $n^{-1/3}$, we query the agents for items that are parts of “large enough” partial matchings. Otherwise, we query everyone at a specific position, and define *simulated values* based on the answers to these queries, ensuring that these values are lower bounds on the corresponding true values. Clearly, the simulated valuation functions are not necessarily unit-sum. For the sake of the presentation, we assume that n is a perfect cube, that is, $n = \alpha^3$ for some $\alpha \in \mathbb{N}$; it is straightforward to extend our analysis to the case where this is not true. See Algorithm 3.

Algorithm 3: FIRSTPOSITIONADAPTIVE (FPA)

All agents are initially *active*.

for every agent i **do**

Query the value v_i^* of i for her top item j_i^* .

if $\max_{i \in N} v_i^* \geq n^{-1/3}$ **then**

for every $\ell \in [n]$ **do**

while there exists a partial matching Z of size $|Z| \geq n^{1/3}/\sqrt{\log n}$, such that an active agent i is matched to an item z_i she ranks at position $\ell' \leq \ell$ **do**

for every agent i in Z **do**

Query the value of i for item z_i

Make i *inactive*

return a matching Y that maximizes the revealed social welfare

else

for every agent i **do**

Query the value u_i of i for the item she ranks at position $n^{1/3} + 1$

for every agent i **do**

Define the simulated valuation function \tilde{v}_i :

- $\tilde{v}_i(j_i^*) := v_i^*$
- $\tilde{v}_i(j) := u_i$ for every item j that i ranks at position $\ell \in \{2, \dots, n^{1/3} + 1\}$
- $\tilde{v}_i(j) := 0$ for every item j that i ranks at position $\ell \in \{n^{1/3} + 2, \dots, n\}$

for every agent i such that $u_i < \frac{1}{2}n^{-1}$ **do**

Set $\tilde{v}_i(j) := \frac{1}{3}n^{-1/3}$ for the item j that i ranks at position $\ell \in \{2, \dots, \frac{1}{4}n^{1/3}\}$.

return a matching $Y \in \arg \max_Z \text{SW}(Z|\tilde{\mathbf{v}})$.

Example 4. Although our running example (see Example 1) is not a unit-sum instance, it is easy to observe that the values of each agent over the items sum up to 45. So, by dividing all values by 45, we do get a unit-sum instance. To avoid having fractional values and be able to directly compare to the optimal social welfare from Example 1, we do not normalize; instead, we appropriately adjust the algorithm instead.

Suppose now that we run FPA on this instance. For each agent i , the algorithm queries and learns the value of i 's most favorite item (here of j_1 for everyone). Then, depending on whether the maximum such value is at least $45 \cdot 8^{-1/3} = 22.5$ (observe the use of the sum of the values 45 here), the algorithm constructs different types of simulated valuation functions. Since $\max_{i \in N} v_i(j_1) = 24 > 22.5$ in our example, the algorithm uses matching-based simulated valuation functions according to the first bullet in its description. The second type of simulated valuation functions is close in spirit to the constructions of λ -TSF and k -FMM, albeit more finely tuned.

For all values of ℓ from 1 to 8, we want to find partial matchings Z_1, Z_2, \dots , each of size at least 2 (since $8^{1/3}/\sqrt{\log 8} \approx 1.15$), which use active agents who rank their matched items

at the ℓ th position or better. For $\ell = 1$, there is no matching of size 2 with two agents that rank their item first, as all agents agree on the best item, and thus only one of them can be matched to it. For $\ell = 2$, a possible matching where everyone considers their item to be at least second best is $\{(i_1, j_2), (i_2, j_1), (i_3, j_3)\}$. Agents i_1, i_2, i_3 become inactive, but there are still feasible matchings on the active agents. A possible second such matching is $\{(i_4, j_1), (i_5, j_3), (i_6, j_2)\}$. Now, agents i_4, i_5, i_6 become inactive, and it is easy to see that $\{(i_7, j_1), (i_8, j_2)\}$ is a third feasible matching. Since there are no more active agents, the algorithm does not look for any matchings for the remaining values of ℓ . The matchings found so far are used to make a second query to the agents about the corresponding items, possibly asking again about j_1 , as is the case for i_2, i_4 and i_7 . Unlike λ -TSF and k -FMM, this part of the algorithm does not try to approximate any values, but rather uses the revealed values along with zeros. The table summarizes the partially revealed valuation profile $\tilde{\mathbf{v}} = (\tilde{v}_i)_{i \in N}$:

| | j_1 | j_2 | j_3 | j_4 | j_5 | j_6 | j_7 | j_8 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| i_1 | 10 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| i_2 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| i_3 | 18 | 0 | 9 | 0 | 0 | 0 | 0 | 0 |
| i_4 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| i_5 | 24 | 0 | 6 | 0 | 0 | 0 | 0 | 0 |
| i_6 | 20 | 7 | 0 | 0 | 0 | 0 | 0 | 0 |
| i_7 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| i_8 | 15 | 10 | 0 | 0 | 0 | 0 | 0 | 0 |

An optimal matching with respect to this $\tilde{\mathbf{v}}$, is $\{(i_1, j_4), (i_2, j_5), (i_3, j_3), (i_4, j_6), (i_5, j_1), (i_6, j_7), (i_7, j_8), (i_8, j_2)\}$ with simulated social welfare 43. The social welfare of this matching with respect to the true valuation profile \mathbf{v} is 66. \square

The distortion of the algorithm is bounded in the following theorem.

Theorem 8. *For unit-sum valuation functions, the distortion of FPA is $O(n^{2/3}\sqrt{\log n})$.*

Proof. Let \mathbf{v} be a valuation profile. Denote by Y the output of the algorithm when given as input the ordinal profile $\succ_{\mathbf{v}}$, and by $X = (x_i)_{i \in N}$ an optimal matching for \mathbf{v} . We consider two main cases, depending on the value $\max_{i \in N} v_i^*$ that the algorithm learns with the first query.

Case 1: $\max_{i \in N} v_i^* \geq n^{-1/3}$. The algorithm makes a second query to an agent i for some item j only if the pair (i, j) is part of a partial matching of size at least $n^{1/3}/\sqrt{\log n}$, involving only active agents, i.e., agents who have not been included in such a partial matching in any previous step. Let Z_1, \dots, Z_λ be all the partial matchings considered throughout the execution of the algorithm. By definition, each such partial matching contains at least $n^{1/3}/\sqrt{\log n}$ agents and an agent is contained in at most one of these matchings. Thus, it holds that $\lambda \leq n^{2/3}\sqrt{\log n}$.

We partition the agents into two sets. The set H contains each agent i who was queried a *second* time about some item z_i she ranks *at least as high* as the item x_i she receives in the optimal matching X . Some agents in H are possibly queried twice for their best item.

The set L contains every agent i who was queried for an item she ranks *lower* than x_i or was queried only *once*. We can write the social welfare of X as

$$\text{SW}(X|\mathbf{v}) = \sum_{i \in H} v_i(x_i) + \sum_{i \in L} v_i(x_i).$$

We will bound each term on the right-hand side separately. For the first term, we have:

$$\sum_{i \in H} v_i(x_i) \leq \sum_{i \in H} v_i(z_i) \leq \sum_{t=1}^{\lambda} \sum_{i \in Z_t} v_i(z_i) \leq \lambda \max_t \sum_{i \in Z_t} v_i(z_i) < n^{2/3} \sqrt{\log n} \cdot \text{SW}(Y|\mathbf{v}).$$

The first inequality holds because $z_i \succ_i x_i$ for every $i \in H$. The second inequality holds because the agents in H are queried only if they are included in one of the partial matchings Z_1, \dots, Z_λ . The last inequality follows from the bound on λ established above, and the fact that $\max_t \sum_{i \in Z_t} v_i(z_i)$ is trivially upper bounded by the social welfare of Y .

To bound the second term, let $Y^{(\ell)}$ be the restriction of Y containing only the agents $i \in L$ for whom x_i is at position ℓ . Observe that all agents in $Y^{(\ell)}$ must be active at the end of the iteration for ℓ . So, it holds that $|Y^{(\ell)}| < n^{1/3}/\sqrt{\log n}$, or else the algorithm would have queried (at least some of) the agents in $Y^{(\ell)}$ for items they value at least as much as x_i before the end of this iteration, contradicting their membership in L . Thus, we get that

$$\sum_{i \in L} v_i(x_i) = \sum_{\ell=1}^n \sum_{i \in Y^{(\ell)}} v_i(x_i) < \sum_{\ell=1}^n \frac{n^{1/3}}{\sqrt{\log n}} \frac{1}{\ell} < \frac{n^{1/3}}{\sqrt{\log n}} 2 \log n = 2n^{1/3} \sqrt{\log n},$$

where the first inequality follows from the unit-sum normalization; in particular, any agent's value for an item at position ℓ is at most $1/\ell$. The second inequality is a simple bound on the harmonic numbers: $\sum_{i=1}^n i^{-1} < 2 \log_2 n$, for $n \geq 2$.

Further, since $\max_{i \in N} v_i^* \geq n^{-1/3}$, we have that $\text{SW}(Y|\mathbf{v}) \geq n^{-1/3}$. Thus,

$$\sum_{i \in L} v_i(x_i) \leq 2n^{2/3} \sqrt{\log n} \cdot \text{SW}(Y|\mathbf{v}).$$

Putting everything together, the distortion of the algorithm in this case is upper bounded by $2n^{2/3} \sqrt{\log n}$.

Case 2: $\max_{i \in N} v_i^* < n^{-1/3}$. We partition the set of agents into two sets, depending on whether their value for the item they rank at position $n^{1/3}+1$ is at most $\frac{1}{2}n^{-1}$. In particular, let $R = \{i \in N : u_i < \frac{1}{2}n^{-1}\}$. We can write the optimal social welfare of X as

$$\text{SW}(X|\mathbf{v}) = \sum_{i \in R} v_i(x_i) + \sum_{i \in N \setminus R} v_i(x_i).$$

We will bound each term separately. For the first term, since $\max_{i \in N} v_i^* < n^{-1/3}$, we clearly have that

$$\sum_{i \in R} v_i(x_i) \leq \max_{i \in N} v_i^* |R| \leq n^{-1/3} |R|.$$

Consider an arbitrary agent $i \in R$ and denote by $j_{i,\ell}$ the item she ranks at position ℓ ; hence, $j_i^* = j_{i,1}$. We will first show that $v_i(j_{i,\frac{1}{4}n^{1/3}}) \geq \frac{1}{3} \cdot n^{-1/3} = \tilde{v}_i(j_{i,\frac{1}{4}n^{1/3}})$. Since $u_i = v_i(j_{i,n^{1/3}+1}) < \frac{1}{2}n^{-1}$, we have that

$$\sum_{\ell=n^{1/3}+1}^n v_i(j_{i,\ell}) \leq (n - n^{1/3} - 1)u_i < \frac{1}{2},$$

and thus, by the unit-sum normalization assumption, we also have that

$$\sum_{\ell=1}^{n^{1/3}} v_i(j_{i,\ell}) \geq \frac{1}{2}.$$

Since $v_i(j_{i,\ell}) \leq v_i(j_{i,1}) < n^{-1/3}$ for every $\ell \in \{1, \dots, \frac{1}{4}n^{1/3} - 1\}$ and $v_i(j_{i,\frac{1}{4}n^{1/3}}) \geq v_i(j_{i,\ell})$ for every $\ell \in \{\frac{1}{4}n^{1/3}, \dots, n^{1/3}\}$, we obtain

$$v_i(j_{i,\frac{1}{4}n^{1/3}}) \geq \frac{\frac{1}{2} - (\frac{1}{4}n^{1/3} - 1)n^{-1/3}}{\frac{3}{4}n^{1/3}} \geq \frac{1}{3}n^{-1/3} = \tilde{v}_i(j_{i,\frac{1}{4}n^{1/3}}).$$

where the second inequality is a matter of simple calculations. So, all the agents in R have value at least $\frac{1}{3}n^{-1/3}$ for the items they rank at positions up to $\frac{1}{4}n^{1/3}$. This implies that the simulated valuation functions, defined by the algorithm, are lower bounds to the real valuation functions.

By Hall's Theorem (Hall, 1935), it is easy to see that there exists a matching of size $\min\{|R|, \frac{1}{4}n^{1/3}\}$ where each agent in R is matched to an item she ranks at the first $\frac{1}{4}n^{1/3}$ positions. Moreover, Y maximizes the social welfare according to the simulated valuation functions. Hence,

$$\text{SW}(Y|\mathbf{v}) \geq \text{SW}(Y|\tilde{\mathbf{v}}) \geq \frac{1}{3}n^{-1/3} \min\left\{|R|, \frac{1}{4}n^{1/3}\right\}.$$

If $|R| < \frac{1}{4}n^{1/3}$, then $\text{SW}(Y|\mathbf{v}) \geq \frac{1}{3}|R|n^{-1/3}$, and thus

$$\sum_{i \in R} v_i(x_i) \leq 3 \cdot \text{SW}(Y|\mathbf{v}).$$

Otherwise, $\text{SW}(Y|\mathbf{v}) \geq 1/12$, and since $|R| \leq n$, we obtain

$$\sum_{i \in R} v_i(x_i) \leq 12n^{2/3} \cdot \text{SW}(Y|\mathbf{v}).$$

For the second term, we further partition $N \setminus R$ into two sets depending on the position of the x_i s. In particular, H is the set of agents $i \in N \setminus R$ who rank x_i at some position $\ell \leq n^{1/3}$, and L is the set of remaining agents $i \in (N \setminus R) \setminus H$ (who rank x_i at some position $\ell > n^{1/3}$). Hence,

$$\sum_{i \in N \setminus R} v_i(x_i) = \sum_{i \in H} v_i(x_i) + \sum_{i \in L} v_i(x_i).$$

First consider the agents in H . Since $\max_{i \in N} v_i^* < n^{-1/3}$,

$$\sum_{i \in H} v_i(x_i) \leq \max_{i \in N} v_i^* |H| < n^{-1/3} |H|.$$

Consider any agent $i \in H$ and any item j that i ranks at some position $\ell \leq n^{1/3}$. Since u_i is the value of i for the item she ranks at position $n^{1/3} + 1$, we clearly have that $v_i(j) \geq u_i = \tilde{v}_i(j) \geq \frac{1}{2}n^{-1}$. Note that there exists a partial matching of size $|H|$ according to which *all* agents of H are matched to items they rank at the first $n^{1/3}$ positions; e.g., the restriction of X on H . Since Y maximizes the social welfare for the simulated valuation functions, we get

$$\text{SW}(Y|\mathbf{v}) \geq \text{SW}(Y|\tilde{\mathbf{v}}) \geq \frac{1}{2}n^{-1}|H|,$$

which immediately implies that

$$\sum_{i \in H} v_i(x_i) \leq 2n^{2/3} \cdot \text{SW}(Y|\mathbf{v}).$$

Finally, consider the agents in L , and distinguish the following two cases depending on the size of L .

- $|L| \leq n^{1/3}$. Since there are at least $n^{1/3}$ different items within the first $n^{1/3}$ positions of each agent in L , by Hall's Theorem, there exists a matching Y' according to which *all* agents in L receive such an item, i.e., every $i \in L$ has (simulated) value at least u_i for the item she gets in Y' . Combining this with the optimality of Y for the simulated valuation functions and the fact that the latter lower bound the real valuation functions, we have

$$\text{SW}(Y|\mathbf{v}) \geq \text{SW}(Y|\tilde{\mathbf{v}}) \geq \text{SW}(Y'|\tilde{\mathbf{v}}) \geq \sum_{i \in L} u_i \geq \sum_{i \in L} v_i(x_i),$$

where the last inequality follows by the definition of L .

- $|L| > n^{1/3}$. Denote by S_L the $|S_L| = n^{1/3}$ agents with the highest values u_i among all the agents in L . We may repeat the above argument for S_L instead of L to get $\text{SW}(Y|\mathbf{v}) \geq \sum_{i \in S_L} u_i$. Then,

$$\text{SW}(Y|\mathbf{v}) \geq n^{1/3} \min_{i \in S_L} u_i \geq n^{1/3} \max_{i \in L \setminus S_L} u_i.$$

On the other hand, we have

$$\begin{aligned} \sum_{i \in L} v_i(x_i) &\leq \sum_{i \in S_L} u_i + (|L| - |S_L|) \max_{i \in L \setminus S_L} u_i \\ &\leq \sum_{i \in S_L} u_i + n \max_{i \in L \setminus S_L} u_i \\ &\leq (1 + n^{2/3}) \cdot \text{SW}(Y|\mathbf{v}). \end{aligned}$$

Therefore, the distortion of the algorithm is at most $16n^{2/3} + 1$ in case 2. Together with case 1, we obtain the desired bound of $O(n^{2/3}\sqrt{\log n})$. \square

6. A General Framework for λ -TSF

In this section we generalize λ -TSF, our algorithm from Section 3, to work for a much broader class of problems, where we are given the ordinal preferences of the agents and access via queries to their cardinal values. We begin with the following general *full information* problem of maximizing an additive objective over a family of combinatorial structures defined on a weighted graph:

Max-on-Graphs: Given a (directed or undirected) weighted graph $G = (U, E, w)$ and a concise description of the set $\mathcal{F} \subseteq 2^E$ of feasible solutions, find a solution

$$S \in \arg \max_{T \in \mathcal{F}} \sum_{e \in T} w(e).$$

Note that One-Sided Matching is a special case; G is the complete bipartite graph on N and A , the weight of an edge $\{i, j\}$ is $v_i(j)$, and \mathcal{F} contains the perfect matchings of G .

What we are really interested in is the social choice analog of Max-on-Graphs where the weights (defined in terms of the valuation functions of the agents) are not given! Instead, we know the ordinal preferences of each agent/node for other nodes (corresponding to items or other agents).

Ordinal-Max-on-Graphs: Here $U = N \cup A$, where N is the set of agents and A is the (possibly empty) set of items; when $A \neq \emptyset$, we assume that G is a bipartite graph with independent sets N, A . Although $G = (U, E)$ is given without the weights, it is assumed that for every $i \in N$ there exists a (private) valuation function $v_i : U \rightarrow \mathbb{R}_{\geq 0}$, so that

$$w(e) = \begin{cases} v_i(j), & \text{if } i \in N, j \in A \text{ and } e = \{i, j\} & \text{(Bipartite agent-item case)} \\ v_i(j) + v_j(i), & \text{if } i, j \in N \text{ and } e = \{i, j\} & \text{(Undirected case)} \\ v_i(j), & \text{if } i, j \in N \text{ and } e = (i, j) & \text{(Directed case)}. \end{cases} \quad (2)$$

We are also given the *ordinal profile* $\succ_{\mathbf{v}} = (\succ_i)_{i \in N}$ induced by $\mathbf{v} = (v_i)_{i \in N}$ and a concise description of the set $\mathcal{F} \subseteq 2^E$ of feasible solutions. The goal is again to find $S \in \arg \max_{T \in \mathcal{F}} \sum_{e \in T} w(e)$.

It is straightforward to extend the notion of distortion (Definition 2) for Ordinal-Max-on-Graphs; assume that $X(\mathbf{v})$ is an optimal feasible solution, $\mathcal{A}_k(\succ_{\mathbf{v}})$ is the feasible solution returned by the algorithm, and take the supremum over all instances of a certain size n .

Notice that for Ordinal-Max-on-Graphs to make sense, \mathcal{F} should be independent of w . For example, if only sets of weight exactly B are feasible, then it is impossible to find even one feasible set without the exact cardinal information in our disposal. Still, it is clear that the above algorithmic problem is very general and captures a huge number of maximization problems on graphs. Of course, not all such problems have a natural interpretation where the vertices are agents with preferences. Before we state the main result of this section, we give several examples that have been studied in the computational social choice literature.

General Graph Matching: Given an undirected weighted graph $G = (U, E, w)$, find a matching of maximum weight, i.e., \mathcal{F} contains the matchings of G . In its social choice analog, $U = N$ and $w(\cdot)$ is defined according to the second branch of (2). A special case of

this problem in which $G = (U_1 \cup U_2, E, w)$ is a bipartite graph, is the celebrated **Two-Sided Matching** problem (Gale & Shapley, 1962; Roth & Sotomayor, 1992).

Two-sided Perfect Matching: A variant of Two-Sided Matching, where $|U_1| = |U_2|$ and only perfect matchings are feasible.

Max k -Sum Clustering: Given an undirected weighted graph $G = (U, E, w)$, where $|U|$ is a multiple of k , partition U into k equal-sized clusters in order to maximize the weight of the edges inside the clusters. That is, \mathcal{F} contains, for each partition of U into k equal-sized clusters, the set of edges that do not cross clusters. This problem generalizes Two-Sided Perfect Matching; see (Anshelevich & Sekar, 2016a). In its social choice analog, $U = N$ and the weights are defined according to the second branch of (2).

General Resource Allocation: Given a bipartite weighted graph $G = (U_1 \cup U_2, E, w)$, assign each node of U_2 to (only) one neighboring node in U_1 so that the total value of the corresponding edges is maximized. There may be additional combinatorial constraints on this assignment, e.g., no more than β_i nodes of U_2 may be assigned to node $i \in U_1$. That is, \mathcal{F} contains the sets of edges that define the partitions of U_2 into $|U_1|$ parts that also satisfy the additional constraints. This problem generalizes One-Sided Matching. In its social choice analog, $U_1 = N$, $U_2 = A$ and $w(\cdot)$ is defined according to the first branch of (2).

Clearing Kidney ℓ -Exchanges: Given a directed weighted graph $G = (U, E, w)$, find a collection of vertex-disjoint cycles of length at most ℓ so that their total weight is maximized; see (Abraham et al., 2007). Here, \mathcal{F} contains the edge set of any such collection of short cycles. In its social choice analog, $U = N$ and $w(\cdot)$ is defined according to the third branch of (2).

We use a variant of λ -TSF, (λ, \mathcal{A}) -TSF, that takes as an additional input an approximation algorithm \mathcal{A} for the problem at hand. There are two main differences from λ -TSF. The simpler one is about the last step; instead of computing a maximum matching, \mathcal{A} is used to compute an (approximately) optimal solution with respect to the simulated valuation functions. The other difference is more subtle. Now we do not want to ask each agent i for her top element of U , but rather for her top element j_i^* that induces an edge included in some feasible solution. That is, j_i^* must be such that there exist $T \in \mathcal{F}$ for which $\{i, j_i^*\} \in T$ (or $(i, j_i^*) \in T$ in the directed case) and any other element with this property is less preferred by i . It is not always trivial to find this element for each agent, but often it can be done in polynomial time; see Corollary 3 for such examples. See Algorithm 4 for a description of (λ, \mathcal{A}) -TSF in pseudocode.

Note that the step of finding the j_i^* s is not given explicitly as it has to be adjusted for the particular problem at hand. As a concrete non-trivial example, consider the perfect matching variant of General Graph Matching, where we only care about *perfect* matchings. In this case, we can check whether an edge $\{i, j\}$ belongs to a perfect matching by removing both i and j and then running the blossom algorithm of Edmonds (1965) on the remaining graph (with all weights set to 1). So, by repeatedly using this subroutine for an agent i starting from her top element and going down her preference list, we can find j_i^* in polynomial time and then make a query for it.

Algorithm 4: (λ, \mathcal{A}) -THRESHOLDSTEPFUNCTION $((\lambda, \mathcal{A})$ -TSF)

// We assume that when $A \neq \emptyset$ (i.e., when there are “items”), $v_i(j) = 0$ for all $i \in A, j \in N$.

Let $\alpha_\ell := r^{-\ell/(\lambda+1)}$ for $\ell \in \{0, \dots, \lambda\}$, where $r = \max_{T \in \mathcal{F}} |T|$

for every agent $i \in N$ **do**

Find i 's favorite $j_i^* \in U$ that defines an edge contained in *some* feasible solution

Query the value v_i^* of i for j_i^*

Let $Q_{i,0} := \{j_i^*\}$

Set $\tilde{v}_i(j_i^*) := \alpha_0 \cdot v_i^* = v_i^*$

for every $\ell \in \{1, \dots, \lambda\}$ **do**

Using binary search, compute the set

$Q_{i,\ell} := \{j \in A : v_i(j) \in [\alpha_\ell \cdot v_i^*, \alpha_{\ell-1} \cdot v_i^*]\}$

Set $\tilde{v}_i(j) := \alpha_\ell \cdot v_i^*$ for every $j \in Q_{i,\ell}$

Let $Q_i := \bigcup_{\ell=0}^{\lambda} Q_{i,\ell}$

Set $\tilde{v}_i(j) := 0$ for every $j \in A \setminus Q_i$

return $\mathcal{A}(\tilde{G})$, where $\tilde{G} = (U, E, \tilde{w})$

For the following theorem, we assume that the optimization problem Π is a special case of Max-on-Graphs with $\max_{T \in \mathcal{F}} |T| = r$. The parameter r allows for a more refined statement; while for the general Max-on-Graphs r may be $\Theta(|U|^2)$, in most cases it is only $O(|U|)$. We further assume that we can efficiently check whether an edge e belongs to a feasible solution. If not, the distortion guarantee of the theorem is still true, but there is no guarantee about the running time of (λ, \mathcal{A}) -TSF.

Theorem 9. *Suppose Π is as described above. If \mathcal{A} is a (polynomial-time) ρ -approximation algorithm for Π in the full information setting, then (λ, \mathcal{A}) -TSF asks $1 + \lambda + \lambda \log r$ queries and achieves distortion at most $3\rho r^{\frac{1}{\lambda+1}}$ for the social choice analog of Π (in polynomial time).*

Proof. Let $X \subseteq E$ be an optimal solution according to the valuation functions v_i , and Z be the solution returned by (λ, \mathcal{A}) -TSF. Also, let Y be an optimal solution with respect to the simulated valuation functions.

In order to unify the notation for the three definitions of edge weights in (1), we write $e = \langle i, j \rangle$ to mean

- (i) $e = \{i, j\}$ with $i \in N, j \in A$ when $A \neq \emptyset$;
- (ii) $e = \{i, j\}$ when $A = \emptyset$ and G is undirected;
- (iii) $e = (i, j)$ when $A = \emptyset$ and G is directed.

Using this notation we can define $X_i = \{j \in U : \langle i, j \rangle \in X\}$, for $i \in N$, and write the optimum as

$$\sum_{e \in X} w(e) = \sum_{\langle i, j \rangle \in X} v_i(j) = \sum_{i \in N} \sum_{j \in X_i \setminus Q_i} v_i(j) + \sum_{i \in N} \sum_{j \in X_i \cap Q_i} v_i(j).$$

We will bound the two terms separately. We begin with the first one:

$$\sum_{i \in N} \sum_{j \in X_i \setminus Q_i} v_i(j) < \alpha_\lambda \sum_{i \in N} \sum_{j \in X_i \setminus Q_i} v_i^* \leq \alpha_\lambda \sum_{\langle i, j \rangle \in X} v_i^* \leq 2r\alpha_\lambda \max_{k \in N} v_k^* \leq 2r\alpha_\lambda \rho \sum_{e \in Z} w(e).$$

The first inequality follows directly by the definition of Q_i . The second inequality follows by extending the scope of the summation to include all (possibly unordered) pairs in X . For the third inequality, it suffices to notice that we simultaneously upper bound the number of terms in the summation by $2|X| \leq 2r$ and each v_i^* by their maximum. Finally, the last inequality follows by the fact that the optimal value with the simulated valuation functions is at least $\max_{k \in N} v_k^*$ and, thus, the solution Z returned by the algorithm achieves at least a ρ -approximation of that.

For the second term we have

$$\sum_{i \in N} \sum_{j \in X_i \cap Q_i} v_i(j) = \sum_{i \in N} \sum_{\ell=0}^k \sum_{j \in X_i \cap Q_{i,\ell}} v_i(j).$$

Now, let us assume that $\lambda > 0$; we will deal with the simpler case where $\lambda = 0$ later. By definition, for any $\ell \in \{1, \dots, \lambda\}$ and any $j \in Q_{i,\ell}$, we have that $v_i(j) \leq \alpha_{\ell-1} \cdot v_i^* = \frac{\alpha_{\ell-1}}{\alpha_\ell} \cdot \alpha_\ell \cdot v_i^* = \tilde{v}_i(j)/\alpha_1$. Also, for $Q_{i,0} = \{j_i^*\}$, we have $v_i(j_i^*) = \tilde{v}_i(j_i^*) \leq \tilde{v}_i(j_i^*)/\alpha_1$. Hence,

$$\begin{aligned} \sum_{i \in N} \sum_{j \in X_i \cap Q_i} v_i(j) &\leq \alpha_1^{-1} \sum_{i \in N} \sum_{j \in X_i \cap Q_i} \tilde{v}_i(j) = \alpha_1^{-1} \sum_{i \in N} \sum_{j \in X_i \cap Q_i} \tilde{v}_i(j) + \alpha_1^{-1} \sum_{i \in N} \sum_{j \in X_i \setminus Q_i} 0 \\ &= \alpha_1^{-1} \sum_{\langle i, j \rangle \in X} \tilde{v}_i(j) = \alpha_1^{-1} \sum_{e \in X} \tilde{w}(e) \leq \alpha_1^{-1} \sum_{e \in Y} \tilde{w}(e) \leq \alpha_1^{-1} \rho \sum_{e \in Z} \tilde{w}(e) \\ &\leq \alpha_1^{-1} \rho \sum_{e \in Z} w(e). \end{aligned} \tag{3}$$

The second inequality follows from the optimality of Y with respect to the simulated valuation functions. The third inequality follows directly from the approximation guarantee of \mathcal{A} : Z attains a ρ approximation of the value achieved by Y . Finally, the last inequality follows from the fact that $\tilde{v}_i(j) \leq v_i(j)$ for every i, j , and thus $\tilde{w}(e) \leq w(e)$ for all $e \in E$.

Now we can put everything together:

$$\sum_{e \in X} w(e) \leq (2r\alpha_\lambda \rho + \alpha_1^{-1} \rho) \sum_{e \in Z} w(e) = 3\rho r^{\frac{1}{\lambda+1}} \sum_{e \in Z} w(e), \tag{4}$$

and this settles the bound on the distortion of (λ, \mathcal{A}) -TSF when $\lambda > 0$.

When $\lambda = 0$, we can repeat the derivation of (3) but without the factor of α_1^{-1} , as this is only needed for the simulated value of items in $Q_{i,\ell}$ for $\ell > 0$ and now these sets are empty. So, $\sum_{i \in N} \sum_{j \in X_i \cap Q_i} v_i(j) \leq \rho \sum_{e \in Z} w(e)$ and then the analog of (4) is

$$\sum_{e \in X} w(e) \leq (2r\alpha_0 \rho + \rho) \sum_{e \in Z} w(e) \leq 3\rho r^{\frac{1}{0+1}} \sum_{e \in Z} w(e).$$

About the running time, it is easy to see that all steps, except running $\mathcal{A}(\tilde{G})$ and finding j_i^* for each $i \in N$, can be done in polynomial time (in particular $O(|U| \log^2 |U|)$). Let $t(|U|)$ be the time needed to check whether an edge can be extended to a feasible solution. Then, finding all the j_i^* s can be done in time $O(|U|^2 t(|U|))$, as we need to perform the check for at most $|U| - 1$ elements per $i \in N$. Hence, if both the feasibility check and \mathcal{A} run in polynomial time, then (λ, \mathcal{A}) -TSF runs in polynomial time as well. \square

For the problems defined above, we can get the following.

Corollary 3. *By choosing \mathcal{A} appropriately, (λ, \mathcal{A}) -TSF asks at most $1 + \lambda + \lambda \log |U|$ queries and achieves distortion at most*

- $3\left(\frac{|U|}{2}\right)^{\frac{1}{\lambda+1}}$ in polynomial time for One-Sided Matching (thus qualitatively retrieving Theorem 2), Two-Sided Matching, General Graph Matching, and Two-Sided Perfect Matching,
- $3|U_2|^{\frac{1}{\lambda+1}}$ for General Resource Allocation;
- $3|U|^{\frac{2}{\lambda+1}}$ for Max k -Sum Clustering;
- $(3\ell - 3 + \varepsilon)|U|^{\frac{1}{\lambda+1}}$ for clearing kidney ℓ -exchanges in polynomial time, for any fixed $\ell \in \mathbb{N}$ and any constant $\varepsilon \in (0, 1)$.

Proof. We begin with One-Sided Matching, Two-Sided Matching, and General Graph Matching. First notice that the size of any matching is at most $|U|/2$ and thus $r \leq |U|/2$. Then, by using an exact algorithm \mathcal{A} for computing maximum weight matchings, such as the blossom algorithm (Edmonds, 1965), Theorem 9 directly implies distortion at most $3(|U|/2)^{\frac{1}{\lambda+1}}$. Regarding the running time, observe that any edge is already a feasible solution, and thus j_i^* is indeed i 's most preferred alternative. Since the blossom algorithm runs in polynomial time, we get that (λ, \mathcal{A}) -TSF runs in polynomial time as well.

For Two-Sided Perfect Matching the argument is as above but one needs to argue about efficiently checking whether a given edge extends to a feasible solution. This, however, is already discussed right after the description of (λ, \mathcal{A}) -TSF.

For General Resource Allocation, we only need to see that an assignment is fully determined by exactly $|U_2|$ edges matching the items to the agents. That is, $r = |U_2|$. Since we do not deal with the running time in this case, we may assume an algorithm \mathcal{A} that solves the full-information problem optimally. Then, Theorem 9 implies distortion at most $3|U_2|^{\frac{1}{\lambda+1}}$. It should be noted here that, depending on the additional constraints imposed by \mathcal{F} , the computation of an assignment may vary from easy (e.g., no constraints) to strongly NP-hard (e.g., the items assigned to each agent should form an independent set in a given graph H on U_2).

For Max k -Sum Clustering, again we do not deal with the running time. Thus, it suffices to use $|U|^2$ as a straightforward upper bound for r and the distortion bound follows.

Finally, for clearing kidney ℓ -exchanges, notice that the number of edges defining a collection of disjoint cycles can be at most $|U|$ and thus $r \leq |U|$. Fix a constant $\varepsilon > 0$. We

can use the polynomial-time $(\ell - 1 + \delta)$ -approximation algorithm of Biró et al. (2009) with $\delta = \varepsilon/3$. Thus, Theorem 9 implies distortion at most

$$3(\ell - 1 + \varepsilon/3)|U|^{\frac{1}{\lambda+1}} = (3\ell - 3 + \varepsilon)|U|^{\frac{1}{\lambda+1}}.$$

Note that the problem is NP-hard, even when $k = 3$ (Abraham et al., 2007). Regarding the running time, we can efficiently check whether an edge (i, j) is in a feasible solution, as it is equivalent to checking whether (i, j) belongs to a cycle of length at most ℓ . For instance, we may find a shortest (j, i) -path in the unweighted version of G ; (i, j) is in a feasible solution if and only if the length of this shortest path is at most $\ell - 1$. Since \mathcal{A} also runs in polynomial time, we get that (λ, \mathcal{A}) -TSF runs in polynomial time as well. \square

7. Conclusion and Open Problems

Our work is the first to study the interplay between elicited information and distortion in One-Sided Matching, as well as more general graph problems, in the context of social choice theory. We have shown several tradeoffs, both in term of possible distortion guarantees, and inapproximability bounds. Our results suggest that using only a small number of queries per agent can lead to significant improvements on the distortion.

As future directions, first it would be very interesting to see if we can come up with algorithms that match the lower bounds of Theorem 6. We managed to do that for the class of k -well-structured instances, but whether it is possible to achieve that for any instance remains to be seen. Perhaps a slightly less ambitious open problem would be to design an algorithm that outperforms the two-queries algorithm presented in Section 5 in terms of the achievable tradeoffs, for agents with unit-sum valuation functions. Another interesting avenue would be to consider randomized algorithms, either in the selection of the matching, or the process of querying the agents, and see if we can obtain significant improvements. Going beyond One-Sided Matching, one could study more general programs, such as those discussed in Section 6 and design tailor-made algorithms with improved tradeoffs between the distortion and the number of queries per agent. Finally, it would be interesting to study a related setting, in which the aim is to achieve the best possible tradeoffs between the distortion and the *total* number of queries, where algorithms could perhaps ask fewer queries for agents that are “easier to please”, leaving more queries for the remaining agents.

Acknowledgments

G. Amanatidis has been partially supported by the NWO Veni project VI.Veni.192.153. G. Birmipas has been supported by the ERC Advanced Grant 788893 AMDROMA “Algorithmic and Mechanism Design Research in Online Markets”, and the MIUR PRIN project ALGADIMAR “Algorithms, Games, and Digital Markets”.

References

Abdulkadiroglu, A., & Sönmez, T. (2013). Matching markets: Theory and practice. *Advances in Economics and Econometrics*, 1, 3–47.

- Abraham, D. J., Blum, A., & Sandholm, T. (2007). Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *Proceedings of the 8th ACM conference on Electronic commerce (EC)*, pp. 295–304.
- Abramowitz, B., & Anshelevich, E. (2018). Utilitarians without utilities: Maximizing social welfare for graph problems using only ordinal preferences. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 894–901.
- Abramowitz, B., Anshelevich, E., & Zhu, W. (2019). Awareness of voter passion greatly improves the distortion of metric social choice. In *Proceedings of the The 15th Conference on Web and Internet Economics (WINE)*, pp. 3–16.
- Adamczyk, M., Grandoni, F., & Mukherjee, J. (2015). Improved approximation algorithms for stochastic matching. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA)*, pp. 1–12.
- Amanatidis, G., Birmpas, G., Filos-Ratsikas, A., & Voudouris, A. A. (2021). Peeking behind the ordinal curtain: Improving distortion via cardinal queries. *Artificial Intelligence*, 296, 103488.
- Anshelevich, E., Bhardwaj, O., Elkind, E., Postl, J., & Skowron, P. (2018). Approximating optimal social choice under metric preferences. *Artificial Intelligence*, 264, 27–51.
- Anshelevich, E., Das, S., & Naamad, Y. (2013). Anarchy, stability, and utopia: creating better matchings. *Autonomous Agents and Multi-Agent Systems*, 26(1), 120–140.
- Anshelevich, E., Filos-Ratsikas, A., Shah, N., & Voudouris, A. A. (2021). Distortion in social choice problems: The first 15 years and beyond. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4294–4301.
- Anshelevich, E., Filos-Ratsikas, A., & Voudouris, A. A. (2022). The distortion of distributed metric social choice. *Artificial Intelligence*, 308, 103713.
- Anshelevich, E., & Postl, J. (2017). Randomized social choice functions under metric preferences. *Journal of Artificial Intelligence Research*, 58, 797–827.
- Anshelevich, E., & Sekar, S. (2016a). Blind, greedy, and random: Algorithms for matching and clustering using only ordinal information. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 390–396.
- Anshelevich, E., & Sekar, S. (2016b). Truthful mechanisms for matching and clustering in an ordinal world. In *Proceedings of the 12th International Conference on Web and Internet Economics (WINE)*, pp. 265–278.
- Anshelevich, E., & Zhu, W. (2017). Tradeoffs between information and ordinal approximation for bipartite matching. In *Proceedings of the 10th International Symposium on Algorithmic Game Theory (SAGT)*, pp. 267–279.
- Bansal, N., Gupta, A., Li, J., Mestre, J., Nagarajan, V., & Rudra, A. (2012). When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4), 733–762.
- Barman, S., & Krishnamurthy, S. K. (2020). Approximation algorithms for maximin fair division. *ACM Transactions on Economics and Computation (TEAC)*, 8(1), 1–28.

- Baveja, A., Chavan, A., Nikiforov, A., Srinivasan, A., & Xu, P. (2018). Improved bounds in stochastic matching and optimization. *Algorithmica*, 80(11), 3225–3252.
- Behnezhad, S., Derakhshan, M., & Hajiaghayi, M. (2020). Stochastic matching with few queries: $(1 - \varepsilon)$ approximation. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 1111–1124.
- Behnezhad, S., Farhadi, A., Hajiaghayi, M., & Reyhani, N. (2019). Stochastic matching with few queries: new algorithms and tools. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2855–2874.
- Benade, G., Nath, S., Procaccia, A. D., & Shah, N. (2017). Preference elicitation for participatory budgeting. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pp. 376–382.
- Bhaskar, U., Dani, V., & Ghosh, A. (2018). Truthful and near-optimal mechanisms for welfare maximization in multi-winner elections. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 925–932.
- Biró, P., Manlove, D. F., & Rizzi, R. (2009). Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discret. Math. Algorithms Appl.*, 1(4), 499–518.
- Blum, A., Dickerson, J. P., Haghtalab, N., Procaccia, A. D., Sandholm, T., & Sharma, A. (2020). Ignorance is almost bliss: Near-optimal stochastic matching with few queries. *Operations research*, 68(1), 16–34.
- Bogomolnaia, A., & Moulin, H. (2001). A new solution to the random assignment problem. *Journal of Economic Theory*, 100, 295–328.
- Boutilier, C., Caragiannis, I., Haber, S., Lu, T., Procaccia, A. D., & Sheffet, O. (2015). Optimal social choice functions: A utilitarian view. *Artificial Intelligence*, 227, 190–213.
- Caragiannis, I., Filos-Ratsikas, A., Frederiksen, S. K. S., Hansen, K. A., & Tan, Z. (2016). Truthful facility assignment with resource augmentation: An exact analysis of serial dictatorship. In *International Conference on Web and Internet Economics (WINE)*, pp. 236–250. Springer.
- Caragiannis, I., Filos-Ratsikas, A., Nath, S., & Voudouris, A. A. (2018). Truthful mechanisms for ownership transfer with expert advice. *CoRR*, abs/1802.01308.
- Caragiannis, I., Nath, S., Procaccia, A. D., & Shah, N. (2017). Subset selection via implicit utilitarian voting. *Journal of Artificial Intelligence Research*, 58, 123–152.
- Caragiannis, I., Shah, N., & Voudouris, A. A. (2022). The metric distortion of multiwinner voting. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*.
- Chen, N., Immorlica, N., Karlin, A. R., Mahdian, M., & Rudra, A. (2009). Approximating matches made in heaven. In *Proceedings of the 36th International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 266–278.
- Costello, K. P., Tetali, P., & Tripathi, P. (2012). Stochastic matching with commitment. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 822–833.

- Edmonds, J. (1965). Paths, trees and flowers. *Canadian Journal of Mathematics*, 17, 449–467.
- Fain, B., Goel, A., Munagala, K., & Prabhu, N. (2019). Random dictators with a random referee: Constant sample complexity mechanisms for social choice. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*.
- Feldman, M., Fiat, A., & Golomb, I. (2016). On voting and facility location. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC)*, pp. 269–286.
- Filos-Ratsikas, A., Frederiksen, S. K. S., & Zhang, J. (2014). Social welfare in one-sided matchings: Random priority and beyond. In *Proceedings of the 7th Symposium of Algorithmic Game Theory (SAGT)*, pp. 1–12.
- Filos-Ratsikas, A., Micha, E., & Voudouris, A. A. (2020). The distortion of distributed voting. *Artificial Intelligence*, 286, 103343.
- Filos-Ratsikas, A., & Miltersen, P. B. (2014). Truthful approximations to range voting. In *Proceedings of the 10th International Conference on Web and Internet Economics (WINE)*, pp. 175–188.
- Filos-Ratsikas, A., & Voudouris, A. A. (2021). Approximate mechanism design for distributed facility location. In *Proceedings of the 14th International Symposium on Algorithmic Game Theory (SAGT)*, pp. 49–63.
- Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1), 9–15.
- Gkatzelis, V., Halpern, D., & Shah, N. (2020). Resolving the optimal metric distortion conjecture. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, p. forthcoming.
- Goel, A., Krishnaswamy, A. K., & Munagala, K. (2017). Metric distortion of social choice rules: Lower bounds and fairness properties. In *Proceedings of the 2017 ACM Conference on Economics and Computation (EC)*, pp. 287–304.
- Goel, G., & Tripathi, P. (2012). Matching with our eyes closed. In *Proceedings of the 53rd IEEE Annual Symposium on Foundations of Computer Science*, pp. 718–727.
- Gross, S., Anshelevich, E., & Xia, L. (2017). Vote until two of you agree: Mechanisms with small distortion and sample complexity. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pp. 544–550.
- Hall, P. (1935). On representatives of subsets. *Journal of the London Mathematical Society*, 1(1), 26–30.
- Hosseini, H., Menon, V., Shah, N., & Sikdar, S. (2021). Necessarily optimal one-sided matchings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, pp. 5481–5488.
- Hylland, A., & Zeckhauser, R. (1979). The efficient allocation of individuals to positions. *Journal of Political economy*, 87(2), 293–314.
- Kempe, D. (2020). Communication, distortion, and randomness in metric voting. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2087–2094.

- Klaus, B., Manlove, D. F., & Rossi, F. (2016). Matching under preferences. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.), *Handbook of Computational Social Choice*, pp. 333–355. Cambridge University Press.
- Kuhn, H. W. (1956). Variants of the Hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3(4), 253–258.
- Mandal, D., Procaccia, A. D., Shah, N., & Woodruff, D. P. (2019). Efficient and thrifty voting by any means necessary. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, pp. 7178–7189.
- Mandal, D., Shah, N., & Woodruff, D. P. (2020). Optimal communication-distortion tradeoff in voting. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pp. 795–813.
- Munagala, K., & Wang, K. (2019). Improved metric distortion for deterministic social choice rules. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC)*, pp. 245–262.
- Plaut, B., & Roughgarden, T. (2020). Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2), 1039–1068.
- Procaccia, A. D., & Rosenschein, J. S. (2006). The distortion of cardinal preferences in voting. In *International Workshop on Cooperative Information Agents (CIA)*, pp. 317–331.
- Roth, A. E., & Sotomayor, M. (1992). Two-sided matching.. Vol. 1 of *Handbook of Game Theory with Economic Applications*, chap. 16, pp. 485 – 541. Elsevier.
- Sönmez, T., & Ünver, M. U. (2011). Matching, allocation, and exchange of discrete resources. In *Handbook of social Economics*, Vol. 1, pp. 781–852.
- Svensson, L.-G. (1999). Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4), 557–567.
- Von Neumann, J., & Morgenstern, O. (1947). *Theory of games and economic behavior*. Princeton University Press.